Automatic Speech Recognition:

A Study of Adaptation Techniques for Noise and Accent Conditions

Proefschrift
ter verkrijging van de graad van doctor
aan de Radboud Universiteit Nijmegen
op gezag van de rector magnificus prof. dr. J.H.J.M. van Krieken,
volgens besluit van het college van decanen
in het openbaar te verdedigen op maandag 13 september 2021
om 14.30 uur precies

door

Yang Sun

geboren op 23 mei 1985

te Peking, China

Promotor:	prof. dr. L.W.J. Boves
Copromotoren:	dr. ir. L.I.J. Cranen dr. L.F.M. ten Bosch
Manuscriptcommissie:	prof. dr. M.A. Larson (Voorzitter) prof. dr. R.W.N.M. van Hout prof. dr. ir. D.A. van Leeuwen prof. dr. ir. H. Van hamme (KU Leuven, België) dr. A.J. van Hessen (Universiteit Twente)

Contents

Chapter 1

Introduction

Automatic speech recognition (ASR) is an application of pattern recognition and machine learning, in which the speech recognizer learns to map acoustic features representing speech to a sequence of words or some other representations of meaning. Especially after the emergence of deep learning algorithms, ASR has become reasonably accurate in quiet environments. However, performance still degrades rapidly when speech is heavily accented or in noisy environments.

It is rather rare to have a noise-free environment, except in laboratories or recording studios. Moreover, the diversity of noises is large. Stationary noises, such as in-car noise, can be separated relatively easily from speech [1], so that their negative impact on ASR performance is not so severe. However, non-stationary noises, such as in restaurants, airports or cocktail parties, are much harder to separate from speech and therefore can cause dramatic degradation of ASR performance [2–4]. Moreover, if the amplitude of the noise is high enough, speakers will tend to raise their vocal effort to enhance the audibility of their voice: the so-call Lombard effect [5–7]. However, that only affects the audibility for human beings. The Lombard effect actually increases the difficulty level of ASR tasks, by introducing a mismatch with normal-volume speech. In short, background noise is one of the most common causes of degradation in ASR performance. Therefore, it is essential to improve noise-robustness of ASR for real-world speech applications.

Besides the countless background noises that influence ASR, recognition performance is also affected by characteristics of individual speakers and speaking habits [8–10], such as gender, age, talking speed, accent, etc. The type and heaviness of accentedness are not only highly influenced by the mother tongue, which is

mostly geographical determined, but also affected by many social factors such as the history and development of cities, education level, etc. Since accented speech poses a practical challenge to ASR systems, a reliable accent representation and identification have immediate applications in robust ASR.

A practical approach to improve both noise and accent robustness of speech recognizers is multi-condition training [11, 12] that involves using training data that contain sufficient variations of noise types or accents. In this approach, the training-test mismatch is reduced by training acoustic models directly on noisy or accented speech signals, rather than only on standard speech from a quiet environment. The training set is selected to reflect the multiple acoustic environments that are considered to be representative for utterances in the target domain. Disadvantages of multi-condition training are twofold: (1) it is difficult to fully predict the range of noise environments and accents which may be encountered in practical applications and (2) the performance for noise-free or accent-free speech deteriorates compared to the baseline models trained with standard clean speech.

In addition to multi-condition training, adaptation is an alternative popular technique to reduce general mismatches between the conditions under which the model was trained and those in recognition. Offline and online adaptation techniques are developed in both the model and feature space [13–15] to achieve a rapid boost of ASR performance with a small amount of in-domain data. In theory, adaptation techniques are expected to asymptotically approach the accuracy of a matched model if sufficient target data is available. Nonetheless, models being adapted with a small adaptation data set suffer from over-fitting and are still vulnerable to unseen noises or multiple noisy conditions at the same time.

Although state-of-the-art ASR systems have been shown to be somewhat robust to various distortions of the input signal or accent variations, even the most powerful systems fall short dramatically when compared with human performance. Under the theme "Bridging the gap between Automatic Speech Recognition (ASR) and Human Speech Recognition (HSR)", the research in this thesis addresses noise and accent robustness to contribute to the improvement of recognition accuracy in both noise and accent robustness by the means of system combination and pronunciation adaptation, respectively. This research was carried out in the Speech Communication with Adaptive LEarning (SCALE) project, part of the Marie-Curie Initial Training Network which was funded by the European Community's Seventh

Framework Programme. Additional research addressing accent robustness was carried out after the end of the SCALE project, as an employee of Nuance™.

1.1 System Combination Techniques towards Noise-robust ASR

System combinations are attractive to noise-robust ASR for two reasons. First, due to the various noise types and noise levels in real applications, it is not trivial to have one single system that is robust against all noisy conditions. System combination is introduced to harness the strengths of different ASR systems which exhibit different error patterns and use multiple knowledge sources to encode complementary information. Second, most of the combination techniques use existing systems which are not required to have extra modifications, so that component systems can be developed independently. In this thesis, system combination techniques are investigated at three stages of processing: early, middle and late.

Early stage combination, usually referring to a concatenation of features from different front ends, is a straightforward way to integrate information [16–19]. For example, short-time cepstral features can be stacked with long-span features that reflect different time resolutions of the same segment of a speech signal and are expected to contain complementary information. Feature concatenation can also embrace so-called secondary features, which are intermediate or final outputs of other systems after certain transformations, if necessary.

Middle stage combination refers to a fusion of multi-stream probabilities or likelihoods that are estimated independently [20, 21]. This approach allows for combining different front ends and enables separate modeling of multiple information sources. This multi-probability-stream approach requires each component system to predict the likelihood vectors of clustered sub-word states based on the same decision tree. Likelihood streams are combined by weighting functions, commonly by means of a SUM or PRODUCT rule, under the basic assumption that all likelihood streams are statistically independent from each other.

Late stage combination can be applied to lattices [22–24] or the best recognition hypotheses [25–27]. The former approach is investigated in this thesis. A lattice is a compact representation of competing hypotheses generated by a decoder that

contains details of both acoustic and language model scores. Usually, the lattice oracle word error rate (WER) or so-called the N-best WER is much lower than the WER based on the single best path, suggesting room of improvement by re-ranking these competing candidates via a system combination.

An essential factor to the success of any combination approach is to have a proper weighting scheme that determines the relative importance of the component systems. Static weight sets may be sufficient for one particular task in a fixed scenario. However, one would prefer to have adaptive weights when there is considerable diversity of the recognition environments. One of the research topics in this thesis is to find a reasonable dynamic weight set function for different types of system fusions, covering all three combination-stage options mentioned above.

1.2 Pronunciation Adaptation Towards Accent-robust ASR

Building effective acoustic models is considerably complicated if these models must operate with a lexicon that only contains canonical phonetic transcriptions. Actual speech often contains pronunciations that deviate from a canonical dictionary. This happens especially in spontaneous conversation [28], when speakers have foreign or domestic accents [29–31] and with dysarthric speakers [32–34]. Accurately modeling pronunciation variability in accented speech, known as *pronunciation modeling* (PM), is an important approach towards obtaining accent robustness. PM consists of detecting and taking into account accent pronunciation variants, using either phoneticians' knowledge or a data-driven procedure [28, 35]. PM is used to include alternative pronunciations from accented origins in the lexicon. Although they are shown to be helpful to improve ASR on specific accented data, phonetic confusion rules usually deteriorate performance on non-accented speech. As this effect becomes more serious when multiple accents can be present at any time, it is important to have a good accent classifier or a good balance of different PM biases for each accent.

1.3 Issues in the Application of Noise- and Accent-robust ASR

Despite several decades of research towards robust ASR, some issues remain that force ASR systems to strike a balance between the requirements posed by multiple accents and background noise. Four of these issues are addressed in this thesis: difficulties in integration of novel robust techniques into a state-of-the-art framework, compromise of recognition performance across a large range of signal-to-noise ratios (SNRs), lack of generalization to multi-stream combinations with real-world data and difficulties in multiple accent robustness in a real-time system.

1.3.1 Integration of Novel Robust Techniques in a State-of-the-art Framework

The *Tandem approach* is a good way to integrate the output of a modeling technique that differs substantially from Gaussian Mixture Models (GMMs). GMMs are widely used because of their simplicity and good representation of many real-world data such as speech features. A good example of alternative features are the posteriors or bottleneck features estimated by a neural network. These so-called secondary features can be used in the Tandem GMM system directly or after some transformations [36, 37]. Logarithmic compression is the most commonly used transformation applied on probability vectors; however, it is not guaranteed to be enough to convert all types of secondary features into Gaussian-like distributions, because there is no constraint on how those secondary features are distributed. It is essential to introduce more general transformations for the Tandem GMMs.

Approach taken in this thesis

In Chapter 2 a Tandem approach is used with the posterior probability estimates from an unconventional exemplar-based classifier, a so-called Sparse Classification (SC) [38, 39]. The motivation is to harness SC's intrinsic noise robustness by using its output posteriors as secondary features in a tandem GMM framework, which can parameterize SC's estimates in order to alleviate the degradation caused by SC's unregularized representation of speech, especially in clean background conditions. Two novel transformations are introduced that aim to 'regularize' the SC's probability vectors. The first approach is a new Gaussianization, with a

special treatment of the non-informative long-tail probability entries, which are replaced by random samples from an artificially generated Gaussian distribution. The goal of using this Gaussian noise is to maximize the probability of the long-tail entries modeled by a dedicated GMM so as to minimize the overlap between non-informative and informative GMM components. The second approach is to apply histogram normalization of the probability vectors, which are forced towards the mean probability distribution while the rank of the individual elements remains unchanged. Histogram normalization happens to both training and test sets, so as to reduce the mismatch of these new secondary features between training and testing data. Additionally, the regularized SC secondary features are stacked with traditional acoustic features Mel-frequency cepstral coefficients (MFCCs) [40] as a feature combination.

1.3.2 Recognition Performances Across a Large Range of SNRs

The reason to combine systems is that the combination can outperform individual systems across multiple conditions on average. However, an issue with the traditional combination approach is that the performance of the combined system in individual conditions often is lower than the performance of the best system for that condition. As a consequence, most previous attempts have combined component systems at similar levels of performances. In this thesis, it is shown that systems with widely different performance can be combined successfully, provided that their contributions can be weighted by the trustworthiness of their output. This is especially important if a large range of signal-to-noise (SNR) must be covered, when some systems outperform others in specific SNR conditions.

Approach taken in this thesis

In Chapter 3 a special technique named Virtual Evidence (VE) is applied to integrate the Sparse Classification (SC) state probability estimates alluded to in Section 1.3.1 into a traditional GMM, which is implemented in a dynamic Bayesian network (DBN) [41–43]. VE is used for combining two feature streams in both training and testing.

To obtain a better understanding of how a dynamic weighting function should be designed, the combination moves to the probability level: two mono-phone posterior

probability streams estimated by a Multi-Layer Perceptron network (MLP) and SC are combined. A new entropy-based confidence estimator is introduced for allocating dynamic combination weight sets to optimize performance in a large range of SNRs. The relationship between entropy of the probability vector and the confidence of the ASR system is estimated independently per component stream in a data-driven fashion. The confidence scores are used as weights after normalization in the combination. Furthermore, a novel arbitration method is proposed to adaptively choose between the SUM or PRODUCT rule in the combination, given the local probability streams.

1.3.3 Generalization to Multi-stream Combinations on Real-world Recordings

The literature lists three shortcomings of combined systems. First, most previous studies of system combination fused similar types of features extracted by means of different front ends [44–46], multi-stream sub-band features [47–49] or multiple types of MLPs [50, 51]. As a result, the combination weight sets, which are used to determine the relative importance among different components, are usually from the same confidence estimator. As a result, the combination is limited in the amount of valuable complementary information it can incorporate. Second, most studies focus on a specific combination method. Ideally, a combination weighting scheme can be generalized to different combination realms, so that more combination approaches can be expected to be evaluated together with the proposed weighting scheme. Third, few studies addressed the combination effect in real-world applications.

Approach taken in this thesis

As an extension of Section 1.3.2, independent confidence models are trained for component systems that use acoustic models (AM)based on deep neural networks (DNN) in Chapter 4. Dedicated confidence models are trained per component system. Up to five different types of DNNs with various backbone neural network techniques produce lattices in parallel, which are then combined offline via confusion network combination (CNC) [52] or minimum Bayes risk (MBR) [23]. The estimated confidences are treated as dynamic combination weights after normalization. Mandarin in-car field data is used for evaluating the effectiveness of this

combination. The data are recordings of end users in various real-world driving conditions, ranging from parked car to highway or country road. It contains both close talk and far talk and the content of speech includes diverse domains such as commands, phone conversations, music, navigation, messaging, virtual assistant, etc.

1.3.4 Performance across Multiple Accents

Pronunciation models (PM) which serve as the bridge between acoustic model (AM) and language model (LM) by mapping phonetic sequences into word tokens. PMs can be utilized to tackle phonetic deviations from a standard pronunciation due to accents and dialects. Their advantage for accent-robustness enhancement is that they do not need accented data for AM training. However, the issue of designing accent-specific sets of phonetic confusions that effectively improve ASR performance is under-investigated. Most previous research focuses on a specific accent, instead of a general overview of most wide-spread accents of the target language [29, 53]. As a result, accent enhancement is accent-specific, so that its effectiveness hinges upon a successful accent classification, which so far has received less attention than language identification.

Approach taken in this thesis

In Chapter 6 an in-depth investigation is conducted into differences between standard and accented Mandarin to identify the phonetic confusions that will make it possible to derive alternative pronunciations for the PM. Three levels of alternative pronunciations are studied from general to fine-grained: (1) global toneless vowel confusions, (2) context-independent vowel or consonant confusions and (3) context-dependent syllable confusions. It is a two-stage development. Stage 1: A data-driven comparison between forced alignment and AM frame-level hypotheses is applied to produce a phonetic confusion matrix, based on which the PM is optimized for each accent. Stage 2: an error analysis of accent-specific enhanced ASR systems is performed to distinguish which confusion pairs improve or degrade ASR results. This analysis outcome leads to a syllable set that optimizes the PM to (1) improve accent-specific performance, (2) boost universal performance across all weak and heavy accents, and (3) make a trade-off between recognition accuracy and Real Time Factor (RTF), as a larger PM leads to a higher RTF.

Additionally, in Chapter 6 accent classifiers are built to further boost the accent robustness.

1.4 Overview of The Chapters

This section contains a short overview of the six chapters in this thesis that report on empirical research.

Chapter 2: Tandem Approach on Top of an Exemplar-Based System towards Noise-robust ASR

Chapter 2 describes a GMM-based tandem system with the probability estimates of an exemplar-based SC system as input – which is transformed into GMM-friendly secondary features via two novel transformations and finally combined with traditional acoustic features in the feature domain. Experiments are initially carried out on the AURORA-2 database, which is a small-vocabulary task consisting of spoken digits artificially corrupted by noise of various types at various noise levels and then generalized to a larger vocabulary task AURORA-4 database, which contains artificially noisified Wall Street Journal utterances.

Chapter 3: Fusion of Parametric and Non-parametric Approaches to Noise-robust ASR

Chapter 3 presents a principled method for the fusion of independent estimates of the state likelihood in a Dynamic Bayesian Network (DBN) by means of the Virtual Evidence (VE) option for improving speech recognition in the AURORA-2 task. A first estimate is derived from a conventional parametric Gaussian Mixture platform; a second estimate is obtained from a non-parametric Sparse Classification (SC) system. During training, the parameters pertaining to the input streams are optimized independently or jointly. The goal is to achieve a universal noise-robust ASR system across a large range of SNR conditions. The chapter is adapted from: Yang Sun, Jort F. Gemmeke, Bert Cranen, Louis ten Bosch, Lou Boves, "Fusion of Parametric and Non-parametric Approaches to Noise-robust ASR", published in Speech Communication 56(1):49–62 · January 2014

Chapter 4: Multi-stream System Combination with Confidence-based Adaptive Weights for Robust ASR

Chapter 4 describes a dynamic combination of SC and MLP. The combination is made in the probability domain and, importantly, a novel trustworthiness-based weight set is estimated independently per stream and associated with the component streams at frame level. In addition, a dynamic switch between SUM and PRODUCT combination rules is investigated. Experiments are carried out on the AURORA-2 database.

Chapter 5: Off-line Lattice Combination with Dynamic Weights on Large Vocabulary Continuous Speech Recognition Tasks

Chapter 5 generalizes the combination work of Article 3 to large vocabulary continuous speech recognition tasks. Also, the two-way combination of probability streams in Article 3 is extended to up to a five-way lattice combination of state-of-the-art ASR systems, with dynamic weights estimated by dedicated neural network-based confidence models. Two lattice combination approaches Confusion Network Combination (CNC) and Minimum Bayes Risk (MBR), are used to compare with static weighting schemes and to each other. Experiments are carried out on Mandarin in-car field test sets.

Chapter 6: Lexicon Study towards Accent-Robust Mandarin ASR

Chapter 6 provides an accent-robustness study for 15 Mandarin accents by pursuing improvement from either the AM or the PM. Under the PM umbrella, tokens with alternative pronunciations are added the lexicon based on phonetic confusions that are either based on linguistic knowledge of the language or fully data-driven. Different levels of phonetic confusions are proposed in order to trade off among multiple accents and between recognition accuracy and speed. Experiments are carried out on Mandarin speech data systematically collected from 15 different geographical regions in China for broad coverage.

Chapter 7: Deep Learning based Mandarin Accent Identification for Accent Robust ASR

Chapter 7 proposes an in-depth study into the classification of regional accents in Mandarin speech. Both Bi-directional Long Short-Term Memory (BLSTM) networks and i-vectors are investigated in an accent classifier of three accent groups via non-metric dimensional scaling (NMDS). As in Article 5, the same accent collection data over 15 regions is used in both of the accent classifier training and evaluation. The chapter is adapted from: Felix Weninger, Yang Sun, Junho Park,

Daniel Willett, “Deep Learning based Mandarin Accent Identification for Accent Robust ASR”, published in Interspeech 2019.

Chapter 2

Tandem Approach on Top of an Exemplar-Based System for Noise-Robust ASR

2.1 Introduction

For more than 30 years Gaussian Mixture Models (GMMs) of Mel-frequency Cepstrum coefficients (MFCC) [54] in Hidden Markov Models (HMMs) have been the favorite means for computing state likelihoods in Automatic Speech Recognition (ASR) [55]. Modeling speech features by means of Gaussian mixtures has proved to be a reasonably powerful approach for clean speech. In noisy conditions, however, the performance of GMM-based classifiers is known to degrade dramatically [56]. The distorted acoustic features of noisy speech signals do not match very well with the statistical distributions derived from clean training material. Multi-condition training is only a partial solution, because it is not feasible to collect training data from all possible future noise conditions. That problem also holds when replacing GMMs with Deep Neural Networks for computing state likelihoods, especially for languages or applications where collecting very large amounts of training data is prohibitive. Therefore, finding new methods that make state likelihood estimation more noise-robust in an insightful manner remains an important problem.

Human listeners are much less affected by additive and/or convolutional noise that even the most advanced ASR systems. Therefore, there has been, and still is, a

keen interest in implementing knowledge about the human auditory system in ASR front ends, e.g. [57–62]. However, none of these approaches have been able to close the gap between human performance and ASR performance. While these approaches model the first stages of the human auditory system, it would seem that a large part of human resilience to noise is due to higher-level neural processes that so far are not accessible for direct observation.

For that reason, researchers have taken recourse to using machine learning approach to 'learn' the features can boost the recognition performance. The use of Multi-Layer Perceptrons (MLPs) for estimating state likelihoods can be considered as an early attempt to develop an alternative to GMMs [63]. A next step towards employing machine learning is the combination of GMMs and MLPs in so-called a Tandem systems [36]. A Tandem system, first introduced in [64], is commonly implemented as a GMM-HMM system which, rather than modeling the distributions of acoustic features directly, takes the posterior or likelihood scores from another classifier as its input features. In combination with MLP classifiers, the Tandem approach was shown to be successful, irrespective of whether final probability estimates of the classifiers, or some intermediate representation such as bottleneck features were used as inputs to the GMM-HMM system [64–68]. Note, however, that in general the distributions of neural network-based classifier outputs are not suited to be modeled by GMMs directly. Typically, several transformations (often designated in short as Gaussianization) must be applied to create features that can be modeled by one or more Gaussian distributions.

Following the successful application of sparse classification (SC) in visual face recognition, this exemplar-based approach was also introduced to the ASR field [69, 70]. SC is a procedure which uses *compressive sensing* or *sparse sampling* to approximate unknown speech segments as a sparse, linear combination of pre-stored speech and noise exemplars from an exemplar dictionary. The procedure does not involve any training process; rather, it is assumed that all relevant variation is represented in one or more exemplars from a pool of dictionary atoms that is constructed by sampling a sufficient number of exemplars from some speech corpus that is representative for the recognition task at hand. For handling noisy speech the dictionary is extended by a number of exemplars of the relevant noise types.

To interface a sparse sampling system with a conventional HMM-based ASR back-end, each exemplar from the speech dictionary is labeled with meta-information about the HMM-state to which it corresponds; noise exemplars are not labeled.

By doing so, it becomes possible to estimate state posterior probabilities of the HMM-states associated with the exemplars by using the weights in the linear combination of the dictionary atoms that were found to reconstruct the incoming speech by the SC procedure. In [70] and [69] the classifiers were made robust against noise by using exemplars with a relatively long duration (e.g. 300 ms), taking advantage of the fact that energy fluctuations related to the noise will often occur at a different time scale than the fluctuations due to movements of the articulators, which are most relevant for the recognition of speech [71]. When the posterior probability estimates from the SC system are fed directly into the search back-end of a conventional HMM decoder, the word accuracies at low SNRs are significantly better than a conventional GMM system using MFCC acoustic features of multi-condition training data [69, 72–74], at the cost however of a significantly worse performance at high SNRs though [69]. Moreover, the long exemplar window in combination with the necessarily large dictionary results in a high computational complexity.

It can be a mutual benefit to marry SC and GMM system. On the one hand, with its power of utilizing long-term temporal information and source separation, SC shows promising performances in very noisy conditions. It could be interesting to investigate if SC's output posteriors can act as a secondary feature that is working in tandem with GMMs. On the other hand, SC does not take advantage of any training algorithms, such as expectation maximization algorithm [75]. Instead, each testing utterance is interpreted via a reconstruction with thousands of independent speech and noise exemplars. In [72] it was shown that a good performance requires a large SC dictionary ($> 8000(speech) + 4000(noise)$) for AURORA-2 [11] task. which implies that, for practical purposes, SC cannot provide an accurate estimate of the likelihood of speech state unless the dictionary pool is large enough to cover all variations in the tests. While the large dictionary pool invariably leads to a relatively high computational complexity, utilizing training processes, such as estimating parameters of GMMs, can benefit SC from smaller or incomplete data.

Figure 2.4 shows that the statistical distributions of MLP-classifier outputs are essentially different from those of SC-classifier outputs. Therefore the Tandem approach as developed for combining outputs from two MLP-based classifiers is not amenable for use with SC-outputs without change. In this chapter, we study the specific requirements that need to be fulfilled in order to be able to successfully employ SC classifier outputs in combination with Neural Network outputs via

a Tandem approach. To that end, we applied two ways for transforming the posterior probability estimates of an SC classifier to secondary features that can be used to train a noise-robust GMM classifier, which will then yield the eventual likelihoods as the input for a Viterbi decoder. Both transformations are designed to take into account the specific distributions of the posterior probability vectors generated by the SC proposed by [69]. The first transformation, first introduced in our previous work [76], is a variant of the log-transform used with the posterior estimates provided by a MLP-based classifier, aiming to facilitate the modeling by means of Gaussians while simultaneously compensating for certain particularities of the classifier that are specific for the SC-approach. The second transformation is reminiscent of Histogram Normalization [77, 78], and aims to mitigate the mismatch between training and test data. We will also explore the effectiveness of both transforms when Histogram Normalization and modified log-transform are combined.

The rest of this chapter is organized as follows: in Section 2.2 the used exemplar-based SC system is briefly reviewed. Subsequently, the principle of the Tandem approach with two modified post-processing methods is described in Section 2.3. The experimental setup and results are given in Section 2.4. This is followed by a discussion in Section 2.5. Finally, the conclusion and future work is described in Section 2.6.

2.2 Sparse Classification (SC) System

2.2.1 Corpus Used for SC Study

The SC classifier used in this study, was first introduced in [79], where it was shown that promising results could be obtained for the well-known AURORA-2 task [11], especially in the lower SNR conditions and for noise types that were represented in the noise dictionary. For the experiments in this chapter, we used the multi-condition training set of AURORA-2 corpus [11] contains 8440 connected digit utterances from the TI-DIGITS database, spoken by 55 male and 55 female speakers. The utterances are artificially corrupted with four noise types (subway, babble, car, and exhibition hall), with SNRs ranging from clean to SNR = 5 dB. In our experiments, we split the multi-condition training set into two parts: we used

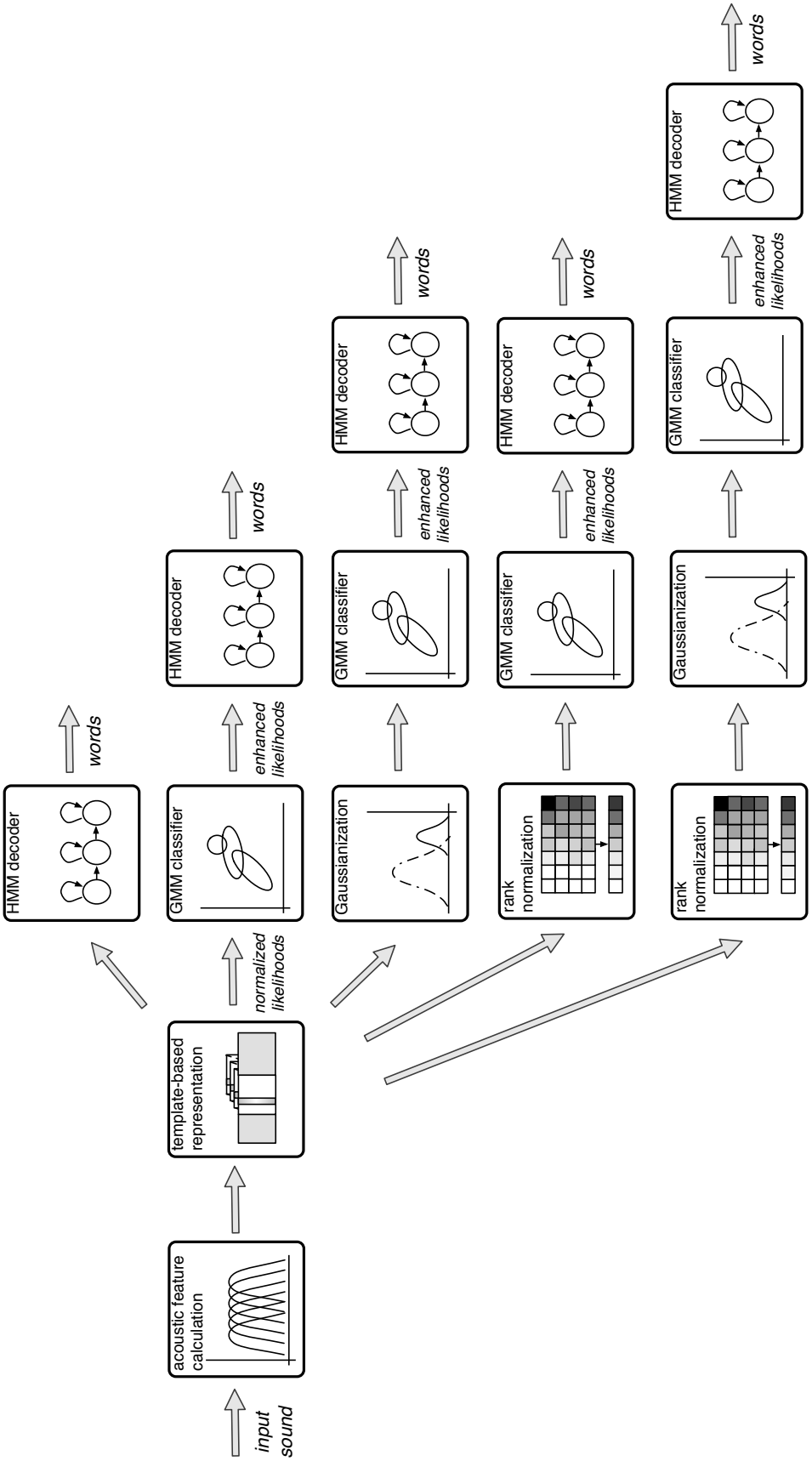


FIGURE 2.1: *Block diagram of the hybrid system (baseline) and different types of Tandem systems proposed in this work.*

7685 utterances for training the Tandem system and the rest 755 utterances as the development data set to optimize the parameters in Eq. 2.6. The split was done in such a way that the speakers in the development set are not present in the training set and the noise types are balanced in both the training and development sets.

For testing, we used test set 'A' (utterances corrupted by the same noise types as in the multi-condition training set) and test set 'B', containing utterances corrupted by four other noise types (viz. restaurant, street, air- port, train station), and which are not included in the noise dictionary employed in the SC system either. Both test set 'A' and 'B' contain 4004 utterances consisting of a sequence of one to seven digits, 1001 utterances for each noise type. All utterances occur in seven noise levels, viz. clean, and SNR = 20, 15, 10, 5, 0, and -5 dB.

2.2.2 State Probability Estimation Using Sparse Representations

In this section, we provide a brief review of the principles of the exemplar-based sparse representation and estimation of class conditional probabilities as proposed in [69]. The SC system approximates the spectrogram Y_w of a noisy speech fragment in a given time window w by the sum of the underlying clean speech spectrogram S_w and the noise spectrogram N_w. Subsequently, it is assumed that both (potentially long) spectrograms S_w and N_w can be sparsely represented by their own linear combination of equally sized spectro-temporal speech exemplars A^s and noise exemplars A^n, respectively. To keep the math tractable, all spectrogram matrices are converted to vectors by vertically stacking the time frames. Thus, representing Y_w, A^s, and A^n and by their vectorized versions y_w, a^s, and a^n, respectively, one obtains:

$$y_w = s_w + n_w = \sum_{j=1}^{J} x_j^s a_j^s + \sum_{k=1}^{K} x_k^n a_k^n \tag{2.1}$$

with x^s and x^n representing the sparse vectors with (non-negative) activation scores for the underlying speech and noise exemplars, respectively. The total number of speech and noise exemplars are denoted by J and K, respectively. The sparse representations can be obtained by minimizing a cost function based on the generalized Kullback-Leibler (KL) divergence [80]. Finding the activations x^s and x^n is the most computational demanding part of the SC system; the time needed

for this part increases linearly with the size of the SC dictionary, consisting of speech and noise exemplars [81].

For all speech exemplars a^s in the dictionary, each time frame $t = 1 \cdots T$ is labeled using state labels $q \in \{1 \dots Q\}$, with Q representing the total number of states. These labels are obtained via forced alignment of the clean speech in the multi-condition training set with a conventional MFCC-based decoder. A $Q \times T$-dimensional binary label matrix $\mathcal{L}_j$ (with only non-zero entries at the {state,frame}-positions that were found by the forced alignment) represents each exemplar in terms of a state sequence. The weight for the j^{th} speech exemplar in approximating an unknown speech segment in window w is denoted by x^s_j. We can then compute an unscaled state likelihood matrix L_w for that speech segment by:

$$\boldsymbol{L}_w = \sum_{j=1}^{J} x^s_{w,j} \mathcal{L}_j \tag{2.2}$$

To obtain state likelihoods for every frame in a speech utterance of arbitrary length, we apply a sliding window, which is visualized in Fig. 2.2. The length of the sliding window T_w equals the length of the exemplars in the dictionary and is shifted over the entire utterance with a step size of $\Delta = 1$ frames. Thus, using eq. (2.2) for each window w, a state likelihood matrix is computed. Subsequently, denoting the columns in the likelihood matrix $\boldsymbol{L}_w$ by $\boldsymbol{l}_{w,\tau}$, where $\tau = 1, \cdots, T_w$ indicates the relative frame position within the window, the state likelihoods for a specific time frame in the utterance are estimated by averaging all state likelihood vectors $\boldsymbol{l}_{w,\tau}$ which pertain to the same absolute position in time:

$$\ell_t = \sum_{\tau=max(1,t-T_{utt}+T_w)}^{min(T_w,t)} l_{t-\tau+1,\tau} \tag{2.3}$$

Typically, in the middle of an utterance the summation boundaries will be from 1 to T_w because T_w overlapping likelihood matrices will be available. Obviously, at the beginning and end of an utterance, fewer state likelihood vectors from overlapping windows are available. The somewhat complicated looking expressions for the summation boundaries in eq. (2.3) are needed to account for this effect.

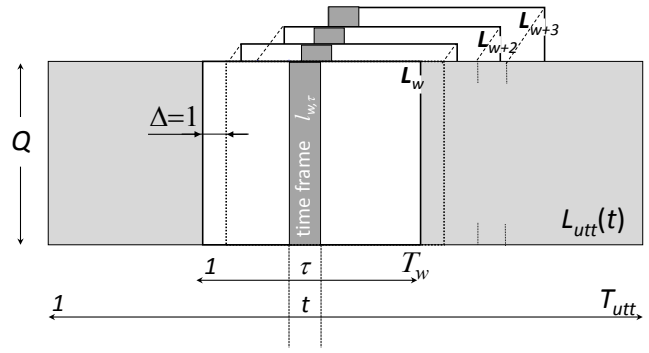


FIGURE 2.2: *State probability estimates for each time frame t in an utterance are obtained by averaging the columns from all state likelihood matrices $L_w(\tau)$ that pertain to an overlapping window and corresponds to the relevant time frame.*

Finally, the posterior probability vector P_t can be obtained by normalizing the likelihood vectors such that the sum of their elements equals one:

$$P_t(q) = \frac{\ell_t(q)}{\sum_{q=1}^{Q} \ell_t(q)} \tag{2.4}$$

2.2.3 Effect of Dictionary Size on the State Posterior Distributions

In [72] it was shown that increasing the size of the speech dictionary has a positive effect on word error rates (WER), albeit at the cost of an increase of the real-time factor (RTF). Here, we show that dictionary size has a strong impact on the crispness (or inversely: the randomness) of the state posterior vectors provided by SC. For that purpose we computed the distributions of the normalized entropy of the 10 ms speech frames in the multi-condition training set, for all SNR conditions in that set as a function of dictionary size. Here, normalized entropy is defined as

$$\tilde{H} = \frac{-\sum_{q=1}^{Q} P_t(q) \cdot log(P_t(q))}{-log(\frac{1}{Q})} \tag{2.5}$$

with $Q = 179$, i.e., the maximum possible entropy if all 179 states would obtain the same posterior probability value. The number of exemplars in the speech dictionaries was $250, 500, 1000, 4000, 8000$, while the size of the noise dictionary remains 4000. The average entropy per frame for the AURORA-2 speech data in

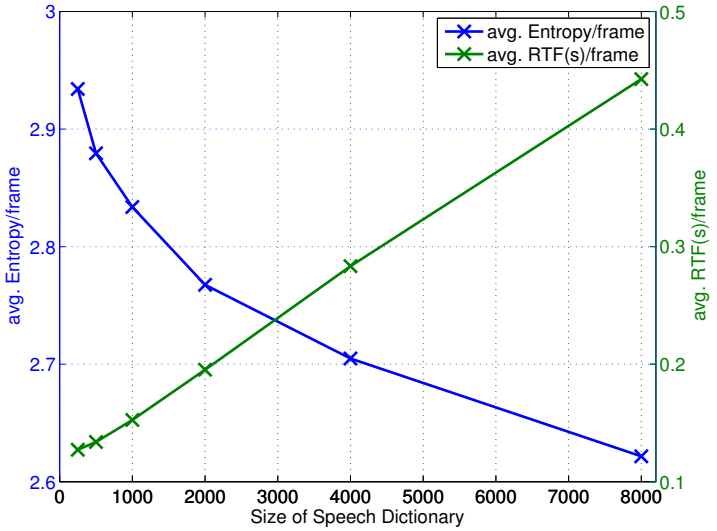


FIGURE 2.3: Averaged frame Entropy of the SC estimated posteriors are plotted in the blue curve for speech dictionary sizes of 250,500,1000,2000,4000 and 8000. Meanwhile, RTFs of the corresponding SC solver are depicted in the green curve.

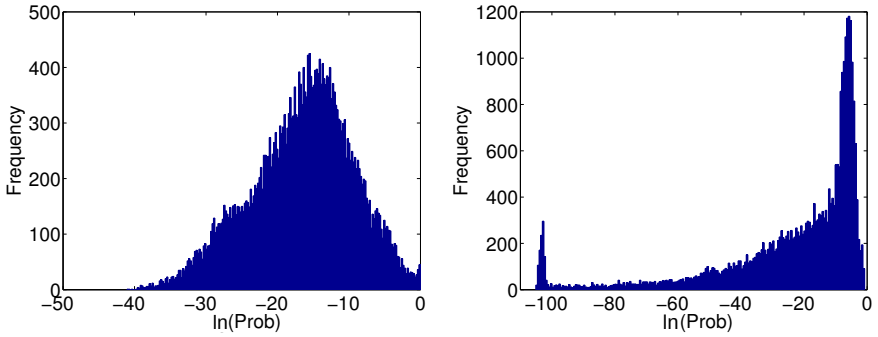
the dev-set, ranging from clean to SNR 5dB, are shown in Figure 2.3. It can be seen that the entropy decreases sharply when the dictionary size increases.

The machine used for the experiments was a AMD Phenom II X4 955, 3.2GHz, with 4GB of RAM. In Figure 2.3, the corresponding averaged RTFs (running time per frame) of the SC solver for each speech dictionary size are depicted in a green curve, showing a linear relationship between the RTF and the size of the speech dictionary.

2.3 Tandem Approach and Feature Processing

2.3.1 The Effect of Logarithmic Transformation

As described in the introduction 2.1, the underlying procedure of the Tandem acoustic modeling approach is to treat the posterior estimates generated by a classifier as *secondary features* that can then be modeled in the same way as MFCC or PLP features, viz. as Gaussian Mixture Models. Since posterior probabilities are constrained to the [0,1] interval, the distributions of their values cannot accurately be represented by means of Gaussian mixtures. Therefore, when state posterior estimates are used as (secondary) features in GMMs it is customary to transform the posteriors in such a way that the resulting distributions are more amenable to



(A) MLP (B) SC

FIGURE 2.4: *Comparing to the histogram of the traditional Tandem MLP feature [11] and the histograms of the SC probability values transformed in a traditional Tandem way (log). The probability entries belongs to the same utterance "3Z82" in the clean condition.*

Gaussian modeling. With posteriors produced by MLP classifiers a logarithmic transform is sufficient to make the distribution more Gaussian-like [64, 65]. A Principal Component Analysis (PCA) can subsequently be used to remove the correlation between the resulting features. Finally, a mean and variance normalization per utterance can be applied to equalize the dynamic range of the data [82].

While MLP classifiers normally generate bimodal probability distributions, i.e. one class takes most of the probability mass (a value close to 1) while the rest of the classes only obtain a small (values close to 0), the SC posteriors distribution usually looks quite different. It often has three distinct modes: (1) values equal to zero (due to the sparseness condition), (2) small values resulting from low activation scores (often reflecting activation of exemplars that served to get a sufficiently accurate acoustic fit of the linear combination, but which are associated with a different state and therefore possibly contain confusion information), and (3) high values from exemplars that take the lion's share of the approximation work and that -hopefully- correspond to the 'true' state.

The difference between the distributions of MLP and SC state posterior estimates is illustrated in Figure 2.4. The two panels show the distributions of the log-transformed posterior probabilities of all 179 states in an utterance *'three', 'zero', 'eight', 'two'* in the clean condition. The utterance comprises 150 frames, so the figures refer to $150 \cdot 179 = 26,850$ data points. From the data in the left hand panel it can be seen that the log-transformed MLP posterior values can be approximated by a small number of Gaussians. However, the data in the right panel, pertaining to the SC-based estimates, can hardly be modeled as a mixture of a small number

of Gaussians. It is doubtful whether a GMM would make any sense at all. The long tail at the left hand side of the distribution of the SC-log-probabilities represents the small values corresponding to the large number of states that obtained small activations. The peak at the left side extreme corresponds to the states that should not be activated at all but still win a tiny bit of weights in the solver because of computer internal numerical floors.

2.3.2 Proposed Gaussianization of SC Posterior Features

It is safe to assume that most of the very small values in the tail down to -100 of Figure 2.4b represent randomly activated and basically ruled-out states. Therefore, we propose the following Gaussianization procedure for the data in the training corpus:

$$y(P_t) = \begin{cases} \ln(P_t) & \text{if } \ln(P_t) > \theta \\ \hat{y} \in \mathcal{N}(\mu, \sigma^2) & \text{otherwise} \end{cases} \tag{2.6}$$

where P_t is the posterior probability vector at time t in Eq. (2.4) and $\hat{y}$ is the probability replacement that is sampled from the artificial Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$.

Rather than treating all elements of the probability vectors from the training set equally, we apply the log-transform only to those entries that are larger than some threshold θ, i.e., the entries that are likely to contain information of the winning states. Entries with values $\leq \theta$ are replaced by samples from a Gaussian distribution with mean of μ and variance of σ^2. In this way, the tiny probability values that are not going to win are replaced by artificial values from a Gaussian distribution, which are guaranteed to be suitable for being modeled by means of a GMM. In testing phase, entries with values $\leq \theta$ are directly replaced by the mean μ to gain the highest likelihoods from the artificial Gaussian distribution.

To assess the possible negative impact of replacing small values in the SC-based probability vectors by artificially constructed values, we first investigated how the WER varies when all probability values below a certain threshold are replaced by a fixed value equal to that threshold. The results are shown in Figure 2.5 and clearly indicate that the recognition performance is not very sensitive to such a procedure as long as the floor value does not exceed 10^{-5}. Since in a Gaussian distribution 97.5% of the samples will fall below the $\mu + 2\sigma$ level, this suggests that replacing $\ln(p) \leq \theta$ values of the training data by some artificial values drawn

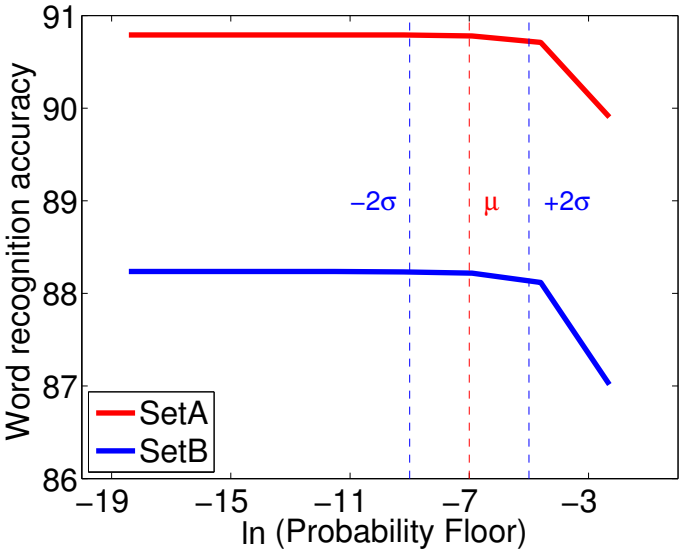


FIGURE 2.5: *Word accuracies for test set 'A'and 'B' obtained with the hybrid SC system when the raw input probability scores are floored. Results shown are average accuracies for the SNR-conditions 0-20 dB. The dashed lines indicate the region where the smallest probability values end up if they are remapped with the Gaussianization procedure used in section 2.4.2.*

from a Gaussian distribution of $\mu = -7$ and $\sigma = 1$ is a safe way to transform the training data, since that operation is not likely to destroy any valuable information in the SC-vectors.

A grid search was used to find the optimal values of the threshold θ, the mean μ and the variance σ^2 by computing the WER on a development set. In order to keep the continuity of the posterior scores, we kept $\theta = \mu$ during the grid search.The search yielded $\theta = \mu = -7$, and $\sigma^2 = 1$ as optimal values.

Figure 2.6 illustrates the effect of the proposed Gaussianization procedure of eq. (2.6) by showing the histogram of probability values for the same utterance as was shown in Figure 2.4b. The red histogram shows all log-prob values above the threshold θ and are retained as is. Note that a small Gaussian-like distribution can be discerned which has its mean located in the pure red area and which represents the larger (more reliable) probability estimates; the more negative values represent the beginning of the long tail that is visible in Figure 2.4b; the blue histogram represents the artificially created Gaussian distribution, which replaces all values in the the long tail below the threshold θ and more or less seamlessly blends in with the left tail values of the red distribution. The values of $\theta = \mu = -7$, and $\sigma^2 = 1$ are the optimized values as reported in [76].

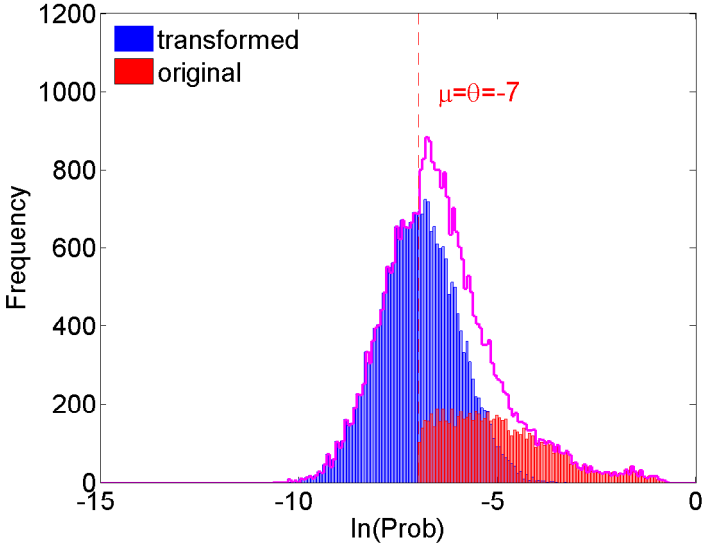


FIGURE 2.6: *Histogram of the Gaussianized probabilities of the same utterance "3Z82" in the clean condition as shown in Fig. 2.4. The histogram in red is based on the original log probabilities that exceed threshold θ). The histogram in blue is based on log-probability entries below the threshold that are transformed according to Eq.* (2.6)

During training, all values below the threshold were substituted by a random number drawn from $\mathcal{N}(\mu, \sigma^2)$ aimed to ensure a proper fit of the data with a Gaussian mixture model. During testing, our goal is different, i.e., we want to ensure that states with very small probability values will not affect the competition between different {frame,state}-path hypotheses in the Viterbi back-end. We tried to accomplish this by replacing all sub-threshold values by a fixed value which was set equal to $\mu = \theta = -7$.

2.3.3 Substituting Posterior Estimates by Their Expected Value Based on Rank

Due to the common problem of ASR acoustic modeling is the variations from system training to the test conditions caused for example by noise contamination, it is hard to bridge the gap while using acoustic features [83], such as MFCC and PLP. Whereas, it seems to be feasible when using the secondary posterior features. In our previous study [84], the SC's probability vectors are ranked, their supports are reduced and finally the N-best states are re-normalized to sum up to 1. Positive results show that the Viterbi decoding of a hybrid system is highly depends on the rank of the probability vectors rather than the absolute value of each probability

entry [85]. Consequently, we proposed a more aggressive transformation approach upon the raw SC probabilities in this work.

As it is shown in Algorithm 1, we first sort all of the training posterior vectors in descending order. Next the expected posterior probability is calculated for every rank. This expected value indicates average behavior of each rank of SC posteriors. In the test phase instead of using the exact averaged posterior vector obtained from the training data, we use rank-based replacement from expected posteriors. In this way, large variations in posteriors are eliminated without changing the probability rank and the numerical range of all of the vectors is normalized such that all of the posterior vectors lie inside a hyper-cube bound by the maximum of expected posterior vector. Finally, the new posterior vector is Gaussianized as in Section 2.3.2.

Algorithm 1 Normalization to the expected value based on rank

(1) At each time frame t in the training data, the probability vector $\boldsymbol{P_t}$ is sorted in descending order as $\{P_t(1)...P_t(k)...P_t(N)\}$, where N is the total number of states and k is the rank of each state probability.

(2) The expected probability at rank k is calculated as

$$P(k)' = \sum_t (P_t(k))/T$$

where T is the total number of frames belonging to the dev-set.

(3) The probability vector at each time frame in the testing data is sorted and replaced by the expected distribution with the state probability rank frozen

$$P_t(k) = P(k)'$$

2.4 Experimental Setup and Results

In this section we assess how the recognition performance of a conventional HMM-decoder back-end varies as a function of the type of modification that is applied to its input vectors, i.e., the vectors with posterior probability estimates from our sparse classification system. To facilitate comparison with performances obtained in previous studies using the SC system (most of them used the AURORA-2 database), we will evaluate our different approaches on AURORA-2 as well.

We compare the five different systems which are schematically depicted in Fig. 2.1. The first system (shown in the top row of Fig. 2.1) is a simple hybrid system that uses the raw posterior vectors from the SC-system directly as its input. On the second row, a Tandem system is shown, in which the posterior probabilities are *not* used directly as local scores, but rather as "secondary features" which are first modeled by Gaussian mixture models (GMMs). The baseline performance of these two systems (shown in the tables at the end of each row) will be used to evaluate the effectiveness of the different strategies in the other three systems. As becomes clear from the figure, these alternative systems differ in the way in which, prior to decoding, the small values in the posterior probability vectors of the SC system were regularized.

On the third row of Fig. 2.1 a Tandem system is depicted where, prior to the Gaussian modeling step, the Gaussianization procedure of Section 2.3.2 is applied. The fourth row also shows a tandem system, but now, instead of replacing the small values of the SC-vectors by values from an artificial Gaussian, the probability vectors of the SC-system are imputed by the rank-based expectation values described in Section 2.3.3. Finally, on the fifth row, a Tandem system is shown in which the rank-based imputation and Gaussianization step are combined.

2.4.1 The Baseline Systems

Three baseline systems are used in this chapter. The first one is the classical GMM-HMM system with 39 dimensional MFCC features as the acoustic input. The second one that are used as point of reference in our experiments are the *hybrid system* depicted in the first row of Fig. 2.1. The last one is the *conventional Tandem system* depicted in the second row of Fig. 2.1. In this section we describe the configuration of these systems and their recognition performance on both test sets 'A' and 'B' of AURORA-2 corpus.

2.4.1.1 Baseline 1: The MFCC-based GMM-HMM

The MFCC input to the GMM-HMM consisted of 39 dimensional vectors containing 12 cepstral features plus a separate log-energy coefficient, as well as the corresponding first and second order delta coefficients. They were based on a 23 band mel frequency spectrum using a frame shift of 10ms and a window length of

25ms. Subsequently, the MFCC coefficients were mean and variance normalized globally. The MFCC feature vectors are represented by diagonal covariance Gaussian Mixtures, which were split once 0.02% convergence of the training likelihood was reached. Our final model consists of 32 diagonal covariance Gaussian Mixtures for each state.

Configuring the HMM-decoder, we followed the standard AURORA-2 approach [11]: Each digit was modeled using a whole-word model consisting of 16 subsequent states; the silence token was modeled by 3 states. Thus there are 179 states to be modeled in total. Additionally, in order to keep the consistency of the back end throughout this chapter, the GMM likelihoods are dumped and imported to a MATLAB implementation of the ASR engine described in [86]. Results can be found in row 'mfcc' of Table 2.1.

2.4.1.2 Baseline 2: The Hybrid SC System

Going from left to right through the top row of Fig. 2.1, also taking into account the sub-systems that the different systems have in common, we discern the following processing blocks:

The acoustic features used, were the same as in [69], i.e., the feature vectors consisted of the magnitudes of 23 Mel-frequency band filters, computed at a rate of 100 frames/s.

In the sparse classifier (described in Section 2.2) different sizes of speech exemplar dictionaries were experimented with. To compose these SC speech exemplar dictionaries, first two speech exemplars per utterance were randomly extracted from the clean-condition training data set of AURORA-2 (in total 2×8440 utterances=16880 exemplars) to compose the SC exemplar training pool. Subsequently, following the setup in [72], subsets of different sizes of 250, 500, 1000, 2000, 4000 and 8000 exemplars were selected randomly from the SC exemplar training pool. Each subset is created from the entire pool independently; in other words, a smaller set was not constructed to form a subset of the bigger ones.

Moreover, a fixed number of 4000 noise exemplars were randomly selected, equally distributed over each noise type in the multi-condition training set. Additionally, again analogous to the approach in [72], 23 artificial noise exemplars were included in the dictionary. Each of the artificial noise exemplars contains a single frequency

band with unit magnitude; the addition of these exemplars are aimed at facilitating the sparse coding of speech that has been contaminated by noise types that are not represented in the noise dictionary and may therefore help to avoid the recruitment of too many unrelated speech exemplars. Recognition performance is summarized in row 'sc prob(hyb)' of Table 2.1.

2.4.1.3 Baseline 3: The Conventional Tandem System

The third baseline system (second row of Fig. 2.1) is a conventional Tandem system, i.e., in addition to the hybrid system it simply contains one extra processing block.

The task of the GMM classifier in the Tandem GMM-HMM system is to convert the 179 dimensional noisy probability estimates of the SC system into a 179 dimensional output vector with less noisy and more reliable state probability estimates. For doing so, each element of the state probability vector is modeled with a GMM consisting of 32 mixture components. The variances of all Gaussians in each mixture were floored to 0.01.

In Section 2.3.1, we described why conventional transformations that are applied successfully in combination with MLP features to make their distributions more Gaussian-like, are unlikely to work equally well with SC posteriors. We experimentally tested this hypothesis by computing the word accuracies for a Tandem system.

The word accuracy scores of the Tandem system in which the SC posterior scores are modeled directly by GMMs as in [82], are shown in the row of 'sc prob(tan)' in Table 2.1.. For test set 'A' the GMM-modeling approach seems to have an advantageous effect, but for the unseen noise types in test set 'B', the GMM modeling approach becomes counter productive for SNRs < 10 dB. On average, the GMM modeling step has not a large effect on recognition performance. Results of applying a log transform (which in the case of MLP features makes the distribution more Gaussian-like) can be found in the row of 'sc log(tan)'. Unfortunately, this transformation makes the word accuracies drop by 5% (absolute) in test set 'A' to 10% in test set 'B'.

An explanation can be found by looking at the histogram of the log-transformed probabilities in Fig. 2.4b. A simple log-transform will cause the HMM decoder to differentiate between the log of small probability values in the same way as it

differentiates between the log of relatively large probability values. However, since in contrast to the MLP system, the low probability values are not considered to be reliable estimates (cf. Sec. 2.2.3), the total cost of the most likely path in the HMM decoder is likely to become too dependent on meaningless probability differences; in deciding which {frame,state}-path is the most likely one, the contribution of the few relatively large probability values (i.e. the ones that *are* considered informative) cannot sufficiently counterbalance the contributions of the large number of small values.

2.4.2 Tandem Log-posteriors with Gaussianization of Small Entries

Under the assumption that the lowest values in the probability vectors of the SC system are effectively noise, the Gaussianization approach discussed in Section 2.3.2 proposes to replace these values by artificially constructed ones that won't upset the subsequent Gaussian modeling stage. In order to investigate what range of values could safely be considered noise, we carried out a pilot experiment in which we investigated to what extent the recognition performance of the hybrid system is susceptible to flooring the smallest values in each probability vector to some fixed value θ. Word accuracies (averaged over SNR conditions between 0 and 20 dB) as a function of θ ($10^{-15} \leq \theta \leq 10^{-1}$) both for test set 'A' and 'B' are shown in Fig. 2.5. This figure suggests that manipulation of elements in the probability vector with values smaller than 10^{-4} can be safely done without affecting word accuracy. Furthermore, the mean μ, standard deviation σ were determined by tuning on a development set. In the end, we applied a threshold $\theta = -7$ to replace all small values in the probability vector by samples from a synthetic Gaussian distribution, whose mean $\mu = -7$ and variance $\sigma^2 = 1$.

The Tandem system, with further Gaussianization of small values in the log-domain, gives the results shown in the row of 'sc Gaus(tan)' in Table 2.1. Such transformation leads to a significant improvement at low SNRs for both test set 'A' and 'B' compared to straightforward modeling the raw probabilities. It suggests that the transformation eases the modeling by GMMs. Besides, the special Gaussianization of small values acts as a regularization (compression) of less important information after the expansion by the log operation.

2.4.3 Rank-based Imputation and Gaussianization of Small Entries

As argued in Section 2.3.3, a rank-based imputation of SC posteriors could be beneficial to avoid a large dispersion of posterior probability values and compensate partially for the mismatch from training to test sets. The expected rank-based probabilities derived from our dev-set are shown in Fig 2.7. For each frame of each utterance in the testing data, the SC posteriors are replaced by this "expected" template without changing the rank of the states. The warped posteriors are then Gaussianized the same way as in Section 2.4.2. Finally, the output "features" are modeled with GMMs.

If we use only the warped posteriors for decoding purpose (without Gaussianization and GMM modeling), it becomes apparent that word recognition performance (row 'sc rank(hyb)' in Table 2.1) degrades but stays in the same ballpark as the raw baseline ('sc prob(hyb)') . This suggests that replacing the posteriors with the expected rank-based value for each time frame without changing the rank does not destroy much of the valuable information in the SC probability vector. This observation underpins the importance of the state rank.

When this regularization is used in the Tandem system (proposed Gaussianization plus GMM modeling), the performance is shown to be improved significantly at low SNRs in both matched and mismatched noise types. The results are shown in the row of 'sc rank(tan)' of Table 2.1.

2.4.4 Modeling the Posteriors Generated by Different Sizes of the SC Dictionary

Previous research has shown that increasing the size of the SC dictionary diminishes the randomness of the SC representation [72], however, at a cost of a dramatic increase of computational complexity. To assess the strength of the proposed Tandem approach, a comparison is made among three systems with different sizes of the SC dictionaries for generating the posterior probability inputs: (1) the hybrid system; (2) the Tandem system with a direct posterior modeling and (3) the Tandem system with both proposed transformations in Section 2.3.2 and 2.3.3. The sizes of the SC's speech dictionary are chosen as 250, 500, 1000, 2000, 4000

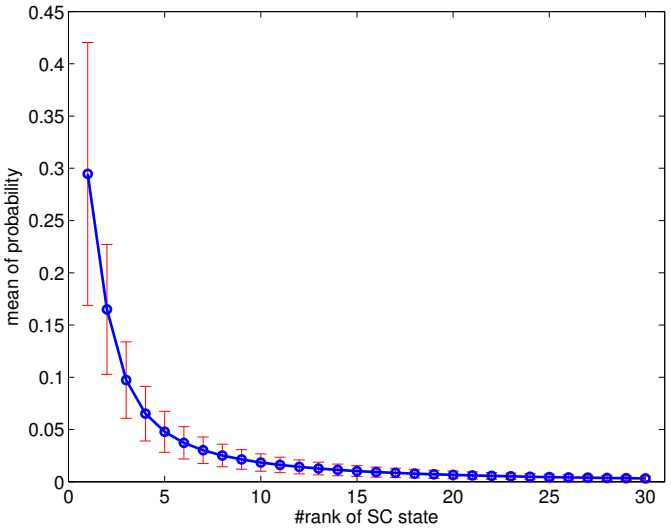


FIGURE 2.7: *The rank-based expected values with the standard deviation of the corresponding rank calculated by averaging all sorted probability vectors in the training data set.*

and 8000, while the size of the noise dictionary was fixed to be 4023 (including 23 artificial noise exemplars). The corresponding performances of the hybrid and Tandem systems are shown in Table 2.2.

The averaged word accuracy from 0 to 20dB in both test set 'A' and 'B' are plotted in Fig. 2.8. Results of the straightforward Tandem system in blue curve shows significant gains comparing to the hybrid one in pink. More substantially with very few speech exemplars, such as 250, the averaged word accuracies of the hybrid system are improved from 69.4% and 67.3% in test set 'A' and 'B' respectively to 90.5% and 84.1% by applying Tandem modeling of the raw SC posteriors ('sc prob(tan)'). These two accuracies are further lifted up to 92.2% and 87.8% when two proposed transformations are adopted. Although the word accuracy is still increasing when a larger SC dictionary is used, Fig. 2.8 shows that the average word accuracy between 0 and 20 dB is convergent since the dictionary size reaches 1000 for both test set A and B.

2.4.5 Combination of MFCC and the Secondary Feature of SC

Previous work [85] studied the combination of MFCC and SC inputs and the interaction between the two. In that scenario, SC's posteriors are used as a 'virtual evidence' for the dynamic Bayesian Network (DBN) which modeling MFCC features

by GMM distributions. The combination effect is promising, nevertheless, a careful tuning is essential to reach an good operating point for diverse SNR conditions. In this work, the transformed SC posteriors are used as a secondary feature combined with MFCC features in a concatenation way. The traditional MFCC features (39 dimensional) described in Section 2.4.1.1 is stacked with the transformed SC secondary features (179 dimensional). Performance of this combination can be found in the row of 'combine' in Table 2.2.

For test set A, a compromise can be found at two ends of the SNR scale. More specifically speaking, the combined system is slightly worse than 'mfcc' baseline at clean and slightly worse than the best SC performance at SNR-5 dB. This can be explained by the fact that the performances of two systems ('mfcc' and 'sc rank(tan)') differ too much in those two extreme conditions. Therefore, a mix of the two provide a mediocre performance. However, looking at the middle range of the SNR plus low SNRs in test set B, the combined system is able to reach at least the better accuracy of the two, if not even better than both. Table 2.2 shows that one can reduce the SC dictionary size by a factor of 32 (from 8000 to 250), meaning approximately 32 times faster, and the averaged word accuracies from 0 to 20 dB will degrade only by absolute 2.1% to 2.3% in test set A and B, respectively. Averaged word accuracy of this combined system is also plotted in Fig. 2.8 as a red curve. For both test set A and B, the combined system achieved a significantly better performance than the best Tandem SC-only system we have ('sc rank(tan)') in green.

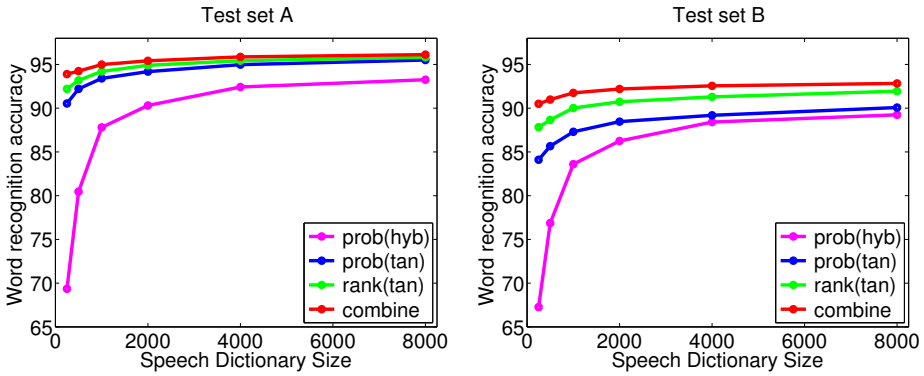


FIGURE 2.8: *Averaged word recognition accuracy (0-20 dB) for test set 'A' and 'B' using different dictionary sizes, corresponding to those four systems in Table 2.2.*

TABLE 2.1: *Word accuracies of all setups in this chapter. "hyb" and "tan" indicate a Hybrid and Tandem system, respectively. 'prob' refers to the raw probability input. 'log' indicates the log transformation of the raw probability vectors. 'Gaus' and 'rank' refer to two proposed transformations: the use of Gaussianization and rank normalization, which are introduced in Section 2.4.2 and 2.4.3, respectively. Note that 'rank' is built upon 'Gaus'. Finally, 'combine' refers to the results of a combined system which stacks inputs of both 'mfcc' and 'rank'.*

		setA							setB						
Word Acc.	clean	20dB	15dB	10dB	5dB	0dB	-5dB	0-20dB	20dB	15dB	10dB	5dB	0dB	-5dB	0-20dB
mfcc	99.34	98.94	98.46	96.96	92.06	71.94	31.52	91.67	98.95	98.46	96.99	92.37	72.98	32.03	91.95
prob(hyb)	97.44	96.98	96.48	95.39	92.61	84.81	65.07	93.25	96.75	95.90	93.64	87.51	72.41	40.15	89.24
prob(tan)	98.90	98.63	98.29	97.41	95.00	88.21	68.09	95.51	98.42	97.49	95.40	88.61	70.47	37.16	90.08
log(tan)	93.80	94.62	93.69	91.92	88.65	80.65	63.73	89.91	91.08	88.83	83.18	73.20	53.39	25.35	77.94
Gaus(tan)	98.68	98.65	98.47	97.65	95.71	89.21	68.56	95.94	98.56	97.89	96.58	90.76	73.63	39.06	91.48
rank(hyb)	96.95	96.46	96.06	94.95	91.75	84.19	65.47	92.68	96.23	95.44	93.09	87.54	72.63	41.33	88.99
rank(tan)	98.72	98.47	98.19	97.47	95.50	89.50	70.75	95.83	98.48	97.74	96.23	91.29	75.99	41.86	91.95
combine	99.14	98.73	98.45	97.70	95.92	89.84	69.52	96.13	98.79	98.17	96.91	92.39	77.89	43.42	92.83

TABLE 2.2: *Word accuracies produced by different sizes of SC dictionaries with four setups. (1) 'prob(hyb)' is the baseline system using raw SC posteriors in a hybrid system; (2) 'prob(tan)' is the direct Tandem modeling approach of the raw SC posteriors; (3) 'rank(tan)' is the system with both of the two transformations proposed in this chapter; (4) 'combine' is the system which stacks MFCC and transformed SC secondary features as its input.*

prob(hyb)		setA							setB						
dict size	clean	20dB	15dB	10dB	5dB	0dB	-5dB	0-20dB	20dB	15dB	10dB	5dB	0dB	-5dB	0-20dB
250	77.22	75.70	74.07	71.66	67.20	58.13	43.21	69.35	76.01	75.15	71.31	64.04	49.80	26.07	67.26
500	87.99	86.89	85.58	82.83	78.33	68.64	51.84	80.45	86.94	85.64	81.60	72.95	57.17	29.38	76.86
1000	93.74	92.96	92.12	90.23	86.30	77.43	58.73	87.81	92.90	91.32	88.55	80.94	64.30	34.32	83.60
2000	95.46	94.79	94.12	92.60	89.30	80.73	61.60	90.31	94.76	93.56	90.96	84.00	67.97	37.15	86.25
4000	96.82	96.16	95.75	94.64	91.89	83.67	64.21	92.42	96.10	95.14	92.89	86.70	71.27	40.03	88.42
8000	97.44	96.98	96.48	95.39	92.61	84.81	65.07	93.25	96.75	95.90	93.64	87.51	72.41	40.15	89.24
prob(tan)		setA							setB						
dict size	clean	20dB	15dB	10dB	5dB	0dB	-5dB	0-20dB	20dB	15dB	10dB	5dB	0dB	-5dB	0-20dB
250	95.33	96.20	95.08	93.76	89.47	78.20	54.36	90.54	95.66	93.88	89.99	80.64	60.35	28.84	84.10
500	96.29	97.07	96.28	95.18	91.36	81.14	57.46	92.21	96.54	95.23	91.79	82.58	62.14	29.44	85.66
1000	97.28	97.50	97.12	96.01	92.69	83.74	61.67	93.41	97.42	96.16	93.10	84.67	65.17	32.61	87.30
2000	98.09	97.91	97.41	96.63	93.68	85.26	64.32	94.18	97.82	96.61	93.81	86.44	67.66	34.52	88.47
4000	98.56	98.30	97.89	96.95	94.45	87.29	66.86	94.98	97.93	97.00	94.30	87.39	69.33	35.85	89.19
8000	98.90	98.63	98.29	97.41	95.00	88.21	68.09	95.51	98.42	97.49	95.40	88.61	70.47	37.16	90.08
rank(tan)		setA							setB						
dict size	clean	20dB	15dB	10dB	5dB	0dB	-5dB	0-20dB	20dB	15dB	10dB	5dB	0dB	-5dB	0-20dB
250	97.21	96.67	96.13	94.63	91.56	82.04	59.24	92.21	96.63	95.70	93.39	86.00	67.44	32.60	87.83
500	97.71	97.33	96.82	95.50	92.38	83.88	62.56	93.18	97.25	96.14	94.25	86.87	68.80	34.11	88.66
1000	98.08	97.74	97.40	96.41	93.64	85.83	64.95	94.20	97.77	96.88	95.10	88.92	71.42	37.00	90.02
2000	98.31	98.13	97.78	96.79	94.60	87.24	67.47	94.91	98.03	97.26	95.56	89.49	73.32	39.55	90.73
4000	98.36	98.19	97.90	97.08	95.17	88.82	69.25	95.43	98.17	97.43	95.84	90.48	74.56	41.40	91.30
8000	98.72	98.47	98.19	97.47	95.50	89.50	70.75	95.83	98.48	97.74	96.23	91.29	75.99	41.86	91.95
combine		setA							setB						
dict size	clean	20dB	15dB	10dB	5dB	0dB	-5dB	0-20dB	20dB	15dB	10dB	5dB	0dB	-5dB	0-20dB
250	98.76	98.28	97.55	96.34	93.29	84.01	59.02	93.89	98.18	97.42	95.71	89.37	71.81	35.72	90.50
500	98.85	98.26	97.69	96.47	93.59	85.18	61.88	94.24	98.34	97.57	96.01	89.73	73.29	37.64	90.99
1000	98.93	98.58	98.05	97.15	94.45	86.72	63.71	94.99	98.50	97.92	96.35	91.11	74.88	39.32	91.75
2000	98.81	98.57	98.22	97.33	95.12	87.84	66.97	95.42	98.56	97.94	96.78	91.51	76.18	42.06	92.19
4000	99.02	98.69	98.33	97.48	95.65	89.16	68.50	95.86	98.70	98.04	96.79	92.14	77.09	42.69	92.55
8000	99.14	98.73	98.45	97.70	95.92	89.84	69.52	96.13	98.79	98.17	96.91	92.39	77.89	43.42	92.83

2.4.6 Large Vocabulary Continuous Speech Recognition – Aurora-4 Corpus

In order to evaluate the effectiveness of proposed secondary-feature transformations and feature combination, we extend the approach to Aurora-4 database (Hirsch, 2001). Like aurora-2, the database is conceived for developing robust frond-ends and speech processing modules to be used in ASR systems, but a larger vocabulary task. It is composed of a 5k word vocabulary based on Wall Street Journal (WSJ0) and contains 3 training sets (with 7138 utterances each one) and 14 test sets (with 166 utterances each one). Several acoustic environments are defined for composing 3 different train sets. Train set 1 consists of clean signals recorded with only one type of microphone, train set 2 (multinoise) contains also additive noise and train set 3 (multi-condition) is composed of signals recorded with different types of microphones and additive noise. Test sets can be bundled by similarity into 4 groups. Group 1 is composed of clean signal (test set 1), Group 2 is composed of noisy signal, contaminated with additive noise (test sets from 2 to 7). Group 3 contains clean signal recorded with a different microphone than train set 1, that is, convolutional noise (test set 8), and Group 4 is composed of noisy signal, with additive noise and convolutional noise (test sets from 9 to 14). In this study, we used 16kHz part of the train set 2 (multinoise) and further split it into training set (7053 utterances) and dev-set (85 utterances) randomly.

2.4.6.1 Sparse Classification Setup

Similar as what is done with aurora-2, two exemplars are randomly extracted from each utterance in training set to compose speech dictionary. Noise dictionary consist of 4000 real noise exemplars from the training data plus 24 artificial ones with identity non-zero values in one and only band per exemplar. All exemplars have a length of 20 frames. Same as aurora-2, KL divergence is used to estimate the sparse representation for each utterance during decoding. For calculating the final probabilities, we attached 41 phoneme labels to each frame in each speech exemplar, therefore the final probability output also has 41 dimensions. Speech and silence amplitude is balanced based on frame error rate (FER) on dev-set.

2.4.6.2 AURORA-4 Experiments

Baseline system uses 39 dimensional MFCC as input features in a GMM-HMM system modeling 15625 triphone states with 4 Gaussian Mixtures each. Results are show in row "mfcc" in Table 2.3. The baseline can reach a performance over 90% at clean, but degrades dramatically under noise. Especially when the audio is contaminated by convoluted noise (set 8 to set 14), the word accuracies get even worse. Averaged word accuracies across test sets with additive noise (set 2 to set 7) is 81.57% and it comes down to 63.66% on average on convoluted noises.

With SC probability output as a secondary feature in the linear and log domain, results are shown in row "prob(tan)" and "log(tan)" respectively. Not like the worse results obtained by directly modeling the probabilities (prob(tan)), the logarithm transformation causes a collapse of the system. This is again caused by the uncontrolled small probabilities are transformed to a large negative value in the log domain, leading to difficulties in modeling. It is worth noticing that the SC system does not yield competitive performance on convoluted noisy test sets (set 8 to set 14). This can be explained by the fact that our training data does not have any such data, and more importantly the algorithm of sparse representation can only separate speech and additive noise in the mel-spectrogram domain. This problem can hardly solved before a convoluted SC algorithm is adopted.

Applying the first transformation – Gaussianization – to regularize the tail of the probability vector (Gaus(tan)), results are improved promisingly comparing to the one without any transformations. And the rank normalization (rank(tan)) can be help to bring a further gain on average, especially in noisy conditions. Although this is not on par with our MFCC-based baseline, it shows that the SC system can still exploit some useful information given the random selection of the SC dictionary pool, and the cursory labels – 41 mono-phones – attached to each frame of each speech exemplar.

A straightforward stacking of 39 dimensional MFCC and transformed (rank normalization plus Gaussianization) 41 dimensional SC leads to a system whose results are given in row "combine". Although now the performance in the same ballpark of the baseline, the combined system is still far behind. This may probably because the performance gap between "mfcc" and "rank(tan)" is too big, thus a compromise is made in the combination. Since mfcc is far better than sc and the dimensions of

the two stream are comparable (39 vs. 41), it is likely that SC obtained too much weight than it should in such combination.

Therefore, we applied principal component analysis (PCA) to the transformed SC stream to reduce its dimension from 41 to 15 and finally combine it with MFCCs. Results of this new combination system is given in row 'comb. (sc15)', where promising results can be found in clean (set 1) or additive noise conditions (set 2 to set 7). The averaged performance over those test sets is improved from 81.57% to 83.30%. In the convoluted part, the new combination still does not redeem the SC system, which must be improved intrinsically before it can contribute anything in the combination.

Nonetheless, this combination shows that SC system indeed can extract useful and compatible information for MFCC. Both of the two proposed transformations are very helpful to fit SC probabilities into GMM building. This gives an evidence of the generality of the proposed approaches on this LVCSR task.

Table 2.3: Word Accuracies of all setups on Aurora-4 corpus.

	additive noise							
	set1	set2	set3	set4	set5	set6	set7	avg.2-7
mfcc	90.02	88.25	83.02	78.01	79.96	82.1	78.05	81.57
prob(tan)	60.41	59.41	51.42	39.93	49.13	45.01	49.54	49.07
log(tan)	1.95	2.36	2.03	0.55	1.62	1.88	1.92	1.73
Gaus(tan)	80.99	78.08	69.83	68.99	72.19	66.22	69.76	70.85
rank(tan)	80.77	77.27	72.04	66.89	71.05	68.84	70.64	71.12
combine	87.11	85.52	78.53	69.06	76.72	75.95	74.81	76.77
comb. (sc15)	90.28	90.02	85.56	77.42	82.95	84.31	79.52	83.30
	convolutional noise							
	set8	set9	set10	set11	set12	set13	set14	set8-14
mfcc	77.9	73.04	62.47	61.18	60.52	65.01	59.71	63.66
prob(tan)	22.84	21.8	19.85	16.54	20.41	17.24	21.95	19.63
log(tan)	1.92	0.81	1.33	-0.77	0.92	1.51	1.47	0.88
Gaus(tan)	44.86	41.8	40.55	42.32	41.99	39.74	42.28	41.45
rank(tan)	47.99	45.01	40.41	45.01	42.87	40.41	42.03	42.62
combine	50.94	48.62	43.06	41.84	44.64	42.69	44.9	44.29
comb. (sc15)	64.79	64.57	58.56	56.8	56.21	58.42	55.54	58.35

TABLE 2.4: *Word accuracies of the Tandem system with proposed Gaussianization and rank-based normalization, comparing to state-of-the-art system from ETSI.*

		test set 'A'		test set 'B'	
	clean	0-20 dB	-5dB	0-20 dB	-5dB
GMM(ETSI)	**99.2**	92.3	43.5	91.8	42.3
baseline sc	97.4	93.3	65.1	89.2	40.2
sc rank (tan)	98.7	95.8	**70.8**	92.0	41.9
combine	99.1	**96.1**	69.5	**92.8**	**43.4**

2.5 Discussion

2.5.1 Comparing to State-of-the-art System

Table 2.4 compares the word accuracy among state-of-the-art front-end processing of [87], the SC baseline, our Tandem system using proposed Gaussianization and rank-based normalization and the combined system with transformed SC scores and traditional MFCC features.

Although there is still an absolute 0.5% drop from ETSI at clean, the absolute improvement of absolute 1.3% of the Tandem approach from the SC baseline certainly bridges the gap. After being combined with MFCC features, the final performance ('combine') reaches 99.1% at clean, which is already in the same ballpark as ETSI. Inheriting the strong robustness at the most noisy condition SNR -5 dB in the match noisy condition (test set 'A'), the Tandem approach with proposed transformations further boosts the performance significantly from 65.1% to 70.8%, leading to an even larger gap comparing to the ETSI system (43.5%) at SNR -5 dB in test set 'A'. The gain at most of the other noisy levels in the match noisy condition (test set A) are also worth to mention: at 0-20dB, the improvement owing to the proposed Tandem modeling is also promising, comparing to the results of ETSI (95.8% vs. 92.3%). On test set B, the 'baseline sc' (hybrid) performs not as well as ETSI. However, 'sc rank(tan)' starts to win across SNR 0-20 dB conditions and the combination outperforms ETSI at SNR -5 dB, besides a further gain over 0-20 dB.

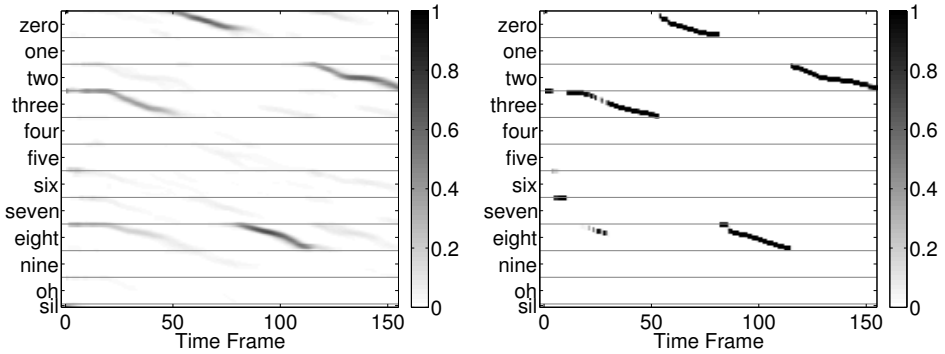
2.5.2 Visualization of the Output Likelihoods of the Proposed Tandem System

Tandem modeling can provide a new representation of the SC posteriors after our proposed transformation pipeline: the rank-based imputation, Gaussianization and GMM-classification. Fig. 2.9 plots the original posteriors and the normalized likelihoods produced by the Tandem system ('rank(tan)') for the same utterance "3Z82" as used in Fig. 2.3, using a dictionary with 8000 speech exemplars. The top row pertains to clean, the second and third row pertain to two different noise types [train noise (from test set 'A') and restaurant noise (from test set 'B'), respectively) at SNR=-5 dB. It turns out that the top posterior traces are "enhanced" so as to be much crisper than the original ones in all conditions. This strengthening is truly beneficial in most of the cases. At clean in Fig. 2.9b, not only the probability shares at each frame are concentrated on a single state, but also a runner-up trace (digit 'eight') from frame 0 to 50 is almost completely eliminated from the competition. The same phenomenon can be observed for noisy data in test set A as well. A more neck-to-neck competition exists between digit 'two' and 'three' of the first digit in Fig. 2.9c, but only the correct one survived the Tandem transformation and classification. This illustrates where the gain comes from after applying the Tandem approach.

As an effect of the proposed Gaussianization, however, the problem caused by the mismatch between training and test data is also amplified. This problem is even more serious in mis-matched noisy conditions, since the output likelihoods of the Tandem system become unreliable for finding good Viterbi paths. For the second digit recognition from frame 50 to 75, a vague but clear trace (digit 'zero') can be observed by naked eyes in Fig. 2.9e. But it disappears in Fig. 2.9f due to the failure of the Tandem GMM classification. This explains why there is a no clear improvement observed at SNR -5 dB in test set B when applying the Gaussianization proposed in Eq. 2.6.

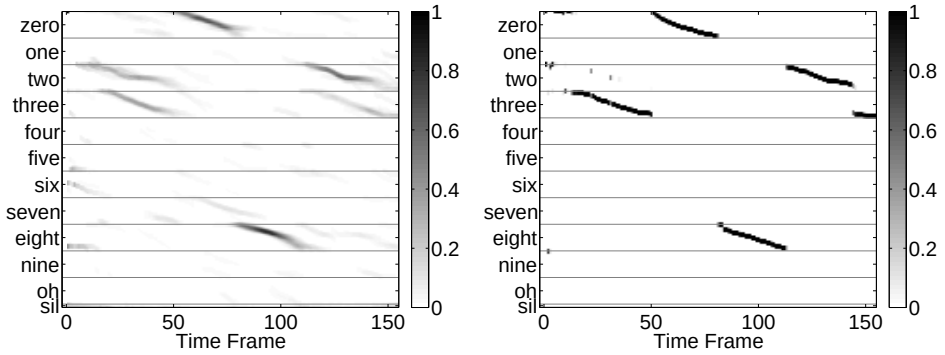
2.5.3 Two Proposed Transformations

Both the Gaussianization and the rank-based warping introduced in this chapter are essential for getting a reasonable Tandem system. Using a high dimensional probability input, simply taking the logarithm will dramatically increase the



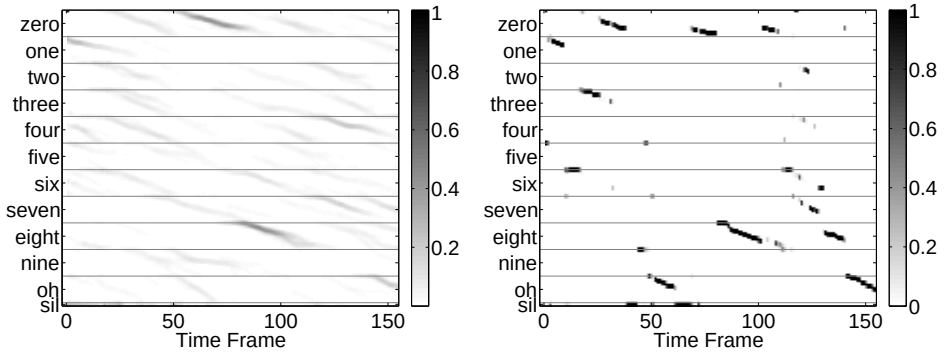
(A) raw prob: clean

(B) tandem: clean



(C) raw prob: train noise -5dB (test A)

(D) tandem: train noise -5dB (test A)



(E) raw prob: restaurant noise -5dB (test B)

(F) tandem: restaurant noise -5dB (test B)

FIGURE 2.9: *Comparison of the raw SC probability matrix for one utterance "3Z82" and the normalized likelihoods by the Tandem approach in different noisy conditions.*

variance of entries with small values. The experiment of the Gaussianization shows that it is important to compress the influence of probabilities with small values while modeling the ones with large values. The experiments of the rank-based warping to the expected posteriors shows: (1) the rank of the states is as important in the hybrid system as the absolute probabilities of each state. When the raw probability vectors at each time frame are replaced by the expected values, the performance of the hybrid system does not differ much. (2) This normalization will help reduce the mismatch between training and test data. The contribution of this normalization is large especially at low SNRs.

Moreover, both transformations are taking advantage of the fact that the format of the raw probability vector is very stable: all entries range between 0 and 1 and sum up to 1. Therefore, it is much easier to transform these probability vector to whatever we need without losing essential information than normal acoustic feature vectors such as MFCC or PLP. Firstly, we artificially replace small entries of our original probability vector with samples from a Gaussian distribution, which eases building GMMs in the training phase. Secondly, we rank-normalized all probabilities as a template probability vector without changing their ranks in the original posterior vector. This approach effectively bridges the gap between training and test data. Neither of these two post-processing methods are based on specific properties of the SC output probability vector, thus it could be possible to extend such two approaches to any other kind of probability vectors for a Tandem modeling.

2.5.4 Varying the Size of the SC Speech Dictionary and Feature Combination

It can be found in Fig. 2.8 that the Tandem modeling improves the performance on average over the hybrid system significantly for all dictionary sizes we tested in this work. The positive effect is more obvious when very few speech exemplars are used. For instance, when we only use 250 speech exemplars, the averaged word accuracy is boosted from 69.4% to 92.2% for test set 'A' and from 67.3% to 87.8% for test set 'B'. Reducing the dictionary size from 8000 to 250 only causes an absolute 3% degradation in the proposed Tandem system, instead of absolute 24%. This finding suggests that a speech reconstruction with only a few speech exemplars can capture the most meaning information of the speech signal already.

Even though the hybrid scores are not satisfying, but after Tandem modeling, the performances are successfully improved to the same ballpark as the system with sufficient exemplars.

Moreover, as the size of the SC dictionary increases, the performance of the hybrid system gets improved significantly. While the improvement of the Tandem system is relatively small – when the size of the speech exemplars is over 1000, the performance is almost stable. It suggests that thanks to the Tandem modeling approach, the requirement of a large dictionary can be much less so that the RTF can be improved linearly according to Fig. 2.3.

The red curve in Fig. 2.3 demonstrates that further improvement can be obtained in both test sets by merging the processed SC stream together with traditional MFCC features. The gain is larger in test set 'B' than 'A', indicating that more complementary information exists between the two for speech with unknown noise types, which is in line with our previous finding [84, 85, 88]. Moreover, the combination improvement is getting more significant if a smaller SC dictionary is used, indicating that the combination can provide larger room for a lighter SC system so that the RTF can be further improved. According to Fig. 2.3, the performance of a combined system (red curve) becomes stable when the the SC dictionary size is larger than 1000. This means that within a combination framework, the SC system can be 8 times faster without a degradation of the word accuracy.

2.6 Conclusion

In this chapter, we attempt to understand how features can be modeled by using a combination of two methods that provide insight into the underlying mechanisms, by applying the Tandem approach on SC posterior scores as a new type of secondary feature in which GMM models are used in the next modeling step. Our goal is to harness the advantage of SC at very low SNRs and use mature GMM techniques to alleviate the intrinsic randomness existing in the SC algorithm. Due to the difficulties of modeling SC scores with a Gaussian or GMMs directly, we proposed two novel transformations. The first one is a Gaussianization with replaced small probability entries with samples from one Gaussian distribution, in order to regularize them. Experimental results show that the modeling can then be

successfully done in the logarithm domain. The second approach, so-called "rank normalization", is aiming to bridge the gap between training and testing data; in another word, to further regularize the SC posteriors. This transformation, applied before the Gaussianization described above, leads to extra improvement according to our experiments, especially on test data in un-observed noise background.

Furthermore, we also explored the performance of the Tandem SC system if the SC dictionary itself is shrunk. More specifically, the size of the SC speech dictionary which is used for the sparse representation is reduced from 8000 to 250 in the end. Thanks to the Tandem approach, we could reach a satisfying performance by using a SC dictionary of 2000 exemplars instead of 8000. Because the computational complexity of the SC system is linear with the size of the dictionary, the reduction of the size of the SC dictionary can largely improve the RTF performance of the system. Additionally, combined with MFCC features by stacking in the feature domain, the gap of WER with an SC dictionary size between 250 and 8000 can be further bridged. Therefore, even better RTF can be gained in the combined system with a reasonable WER.

In theory, both transformations should fit for Gaussianizing any kind of posterior vectors. As the next step, it would be interesting to verify if the proposed Gaussianization is effective on other kinds of secondary features. Secondly, the dimension of 179 input features can be less economic than what is normally used for most of the GMM-HMM systems, whose input size is usually between 30 to 60. Thus, modeling phoneme posteriors instead of state posteriors may be one idea to explore.

In the next chapter, we would further investigate a deeper fusion of non-parametric SC system and parametric GMM system with and without a joint training. Instead of feature concatenation, information from the SC system will be imported to the GMM architecture on the platform called Dynamic Bayesian Network. The relative importance of the streams will be adjusted by various weights. The focus of the next chapter is how to achieve a good combination effect across a large range of SNRs.

Chapter 3

Fusion of Parametric and Non-parametric Approaches to Noise-robust ASR

3.1 Introduction

Parametric models, such as Gaussian Mixture Models (GMMs), have been used successfully in a wide range of pattern recognition problems. For example, acoustic models based on GMMs of Mel-frequency Cepstrum coefficients (MFCC) in Hidden Markov Models (HMMs) have dominated Automatic Speech Recognition (ASR) for the last 30 years [55]. Modeling speech features as Gaussian mixtures has proved to be a powerful approach for clean speech. In noisy conditions, however, the performance of GMM-based recognizers is known to degrade dramatically. Basically, this is because it is difficult and expensive to model speech in noise sufficiently accurately using GMMs if one wants to account for all potentially relevant (non-stationary) noises. As a consequence, the parameters characterizing observed noisy speech signals often do not match the distributions derived from the training material which has been recorded in noise-free conditions or in conditions with only a small number of noise types.

In the past, several different approaches have been proposed to make GMM-based HMM systems more robust against noises that were not represented in the training data. One approach, exemplified by [87], consists of trying to remove the noise

from the signal. By doing so, the mismatch between the trained distributions and the observed signal is reduced. Another approach, which comes in several different flavors, is known as *Missing Data Theory* [89–91]. Basically, this class of approaches aims to determine the acoustic features that are not dominated by noise, and to base decoding on that subset of the features. Yet another set of approaches, known as *model compensation*, exemplified by [92], aim at adapting the trained distributions to the characteristics of the noise. All approaches mentioned above have in common that they can improve recognition performance in signal to noise ratios (SNR) between 20 and 0 dB substantially, although mostly at the cost of some degradation of the performance in clean conditions.

Recently, a new approach, named Sparse Classification (SC) [69], was introduced to the ASR field, which holds the promise of producing robust estimates of the posterior probabilities of phones or states, even in SNR$<0\,$dB conditions. Because SC makes no assumptions about the distributions of the acoustic features, nor of the shapes of the classes and the boundaries between these, the new approach can be regarded as non-parametric. Using a dictionary of speech and noise segments, called *exemplars*, represented in the form of Mel-scaled magnitude spectrograms, clean and noisy speech can be approximated as a linear combination of a small number of such exemplars. By only using the linear combination of the *speech* exemplars in the approximation as a basis for decoding (and discarding the selected noise exemplars), it is possible to improve recognition performance in the lower SNR conditions, even in the $-5\,$dB condition. However, for clean speech the performance of this (non-parametric) SC approach falls well below the best conventional (parametric) GMM-based systems.

In this chapter, we investigate a dual-input ASR system that fuses the state likelihoods obtained from parametric GMMs and the posterior probabilities from a non-parametric SC system, in such a manner that the dual-input system can harness the power of the GMMs for accurately modeling speech in clean conditions, and at the same time profit from the performance of the SC system in noisy conditions.

Multiple methods for combining information streams in ASR have been proposed. For instance, there have been several attempts to augment acoustic features such as MFCC or PLP coefficients by different types of information that can be derived from the speech signals, such as articulatory features [93] or state posterior probabilities computed by means of MLPs or SVM-based classifiers [94–97]. These approaches

have in common that they *append* the additional features to the original acoustic features, or that they use the alternative features *instead of* the original acoustic features. More recently, Conditional Random Field approaches have been used for merging evidence from qualitatively different sources [98]. Besides the early fusion approaches mentioned above, there are also late fusion approaches, such as ROVER [99], which fuse the *output* of multiple independent recognition systems.

Our approach is similar to the fusion of probabilities at the HMM-state level applied in studies such as [100–105]. However, rather than trying to find optimal procedures for combining independent state posterior probability estimates obtained from SVM, MLP and GMM systems during *decoding* (e.g., by weighted multiplication or addition), we explore whether the concept of Virtual Evidence (VE) in a Dynamic Bayesian Network (DBN) [106] may bring an additional advantage. The VE-concept in DBNs provides a mathematically coherent framework that makes it possible to *jointly train* all parameters of the DBN, such as GMMs and Conditional Probability Tables (CPTs). Moreover, the VE-concept is not limited to combining feature streams at the state level, but makes it possible to insert external evidence at all levels in a network. However, before attempting fusion above the state level, we first want to fully understand the fundamental issues related to this approach to fusion at the state level.

In previous papers we used a trial-and-error approach for finding the best way to combine the evidence from GMMs and an SC-system. In [107] we started exploring the effect of a weighted combination of GMM and SC streams. In [108] the SC stream was represented in the form of posterior probabilities of all HMM-states, and in [84] we investigated the impact of keeping only the most likely HMM-states provided by SC as Virtual Evidence. In this chapter we develop a framework that unifies our earlier experiments, and that provides a principled understanding of how the optimal weights of the streams are determined by the distributions of the input streams. This framework not only allows us to explain and interpret the commonalities and differences in our earlier experiments, it is also instrumental in setting directions for future research.

The long-term goal of our research is to improve the noise-robustness of ASR systems. As a first step in that direction, we investigate the recognition performance that can be obtained in the AURORA-2 connected digit recognition task [11]. For our research we used the Graphical Modeling Toolkit (GMTK) [109], because GMTK provides a flexible platform to investigate the use of VE in a DBN. More particularly, GMTK

provides easy access to all GMMs and CPTs that are formed during training. We use this feature to systematically investigate the degree to which different model components and different training scenarios affect the decoding results.

The rest of the chapter is organized as follows. In Section 3.2 we review the basics of the SC and DBN systems and introduce our dual-input DBN, followed by a description of the experimental settings in Section 3.3. We report and discuss the results of our experiments on AURORA-2 in Section 3.4. Conclusions and suggestions for future work are presented in Section 3.5.

3.2 Model Description

In this chapter, we use a dual-input DBN to fuse likelihoods obtained from GMMs with the state posterior probability estimates provided by the SC system described in [69]. In Section 3.2.1 we first summarize how the estimates from the SC system are obtained. Subsequently, in Section 3.2.2, we describe the DBN in more detail.

3.2.1 State Probability Estimation Using Sparse Classification

In our SC approach audio signals (speech as well as noise) are represented in the form of the magnitudes of 23 band-pass filters, equally spaced on a Mel-frequency scale, and sampled at 100 frames/s. For the experiments in this chapter each frame of the clean speech in the AURORA-2 training database was labeled with the state-id it pertained to. The state labels were obtained by means of a forced alignment, using a conventional HMM system with 16-state word models for the eleven digit words, a 3-state silence model and a 1-state short pause model (identical to the middle state of the silence model), 179 states in total. Subsequently, 4000 segments of Mel-frequency spectrograms with a duration of 30 frames (=300 ms) were randomly extracted from the clean training speech and stored in a so-called *exemplar dictionary*. In addition, for each exemplar the corresponding sequence of thirty HMM state labels is stored.

The noise used to corrupt the speech in the multi-condition training database was reconstructed by subtracting the clean speech from the corresponding noisified

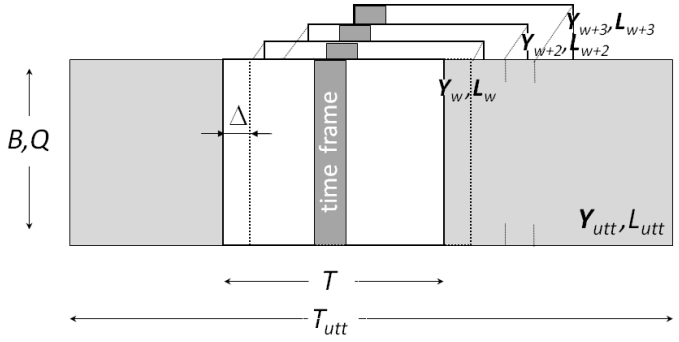


FIGURE 3.1: *State probability estimation by SC: First, an utterance of arbitrary length T_{utt} is represented by shifted spectrogram windows $\boldsymbol{Y}_w$ with duration T frames and B Mel-frequency bands. Δ denotes the window shift. For each window w, SC yields a state likelihood matrix $\boldsymbol{L}_w$. The Q state probability estimates for each time frame, visualized by the dark gray column, are computed by summing (and normalizing) the corresponding columns of all overlapping likelihood matrices.*

speech. Then, 4000 noise segments, each with a duration of 30 frames (=300 ms), were randomly selected from the noise signals and added (without state annotation) to the exemplar dictionary.

SC works by first representing an utterance in the form of overlapping windows spanning $T = 30$ frames (= 300 ms), denoted by $\boldsymbol{Y}_w$ with w the window index (see Fig 3.1). Each spectrogram window is represented as a sparse, non-negative linear combination of atoms in the exemplar dictionary by minimizing the Kullback-Leibler divergence between the linear combination and the observation, regularized using a sparsity-inducing L-1 norm of the exemplar weights [69]. We assume that the speech exemplar weights needed for representing the spectrogram $\boldsymbol{Y}_w$ (the weights of the noise exemplars are discarded) are also meaningful for describing the relative likelihood of the corresponding state labels. This enables us to use the speech exemplar weights together with the stored exemplar-state mapping to calculate the likelihood of all states for each frame in the window as a weighted sum of state occupancies. This yields a $Q \times T$ dimensional state likelihood matrix $\boldsymbol{L}_w$ for each window, with $Q = 179$ states.

Since we apply a sliding window approach, each frame in the utterance is associated with multiple overlapping state likelihood matrices (cf. Fig 3.1). By summing (and normalizing) the relevant columns of the state likelihood matrices, we obtain the state posterior probability estimate denoted $p(q_t|SC_t)$, a 179 dimensional vector at every 10ms frame in which each component corresponds to a probability estimate

for each state q_t used in the 16-state word models. For a more in-depth explanation we refer to [69].

3.2.2 Dynamic Bayesian Networks

Dynamic Bayesian Networks (DBN) are a subset of graphical models that encompass many existing algorithms for ASR [110]. The DBN framework allows one to make explicit assumptions about relationships between variables in a model that are difficult to express in a conventional HMM. This greatly facilitates the extension of existing models and, more importantly, the exploration of novel ideas [111, 112].

3.2.2.1 DBN Baseline

The DBN baseline architecture used in this study is taken from the AURORA-2 tutorial[1] that comes with the GMTK distribution [109]. Denoting the sequence of values that a variable assumes in subsequent frames t during the interval $[1, T]$ as $(\cdot)_{1:T}$, the single input DBN (of which three of the T frames are depicted in Fig. 3.2) complies with the following algebraic factorization of the joint probability:

$$\begin{aligned} & p(w_{1:T}, w^{tr}_{1:T}, w^{ps}_{1:T}, q^{tr}_{1:T}, q_{1:T}, y_{1:T}) = \\ & \prod_{t=1}^{T} \{ p(y_t|q_t) f(q_t|w^{ps}_t, w_t) f(w^{tr}_t|w^{ps}_t, w_t, q^{tr}_t) \\ & p(q^{tr}_t|q_t) f(w^{ps}_1) p(w_1) \} \prod_{t=2}^{T} \{ p(w_t|w^{tr}_{t-1}, w_{t-1}) \\ & f(w^{ps}_t|q^{tr}_{t-1}, w^{ps}_{t-1}, w^{tr}_{t-1}) \} \end{aligned} \tag{3.1}$$

in which y_t is the acoustic observation at time t, $f(\cdot)$ indicates deterministic CPTs, $p(y_t|q_t)$ represents a continuous probability density function, and the other factors $p(\cdot)$ represent discrete density CPTs.

As in [113], the variable w (cardinality 13) represents a linguistic "word" unit (11 digits 'zero' to 'nine' and 'oh'), 'silence' or 'short pause'; w^{ps} (cardinality 16, 3 or 1 for digits, silence and short pause, respectively) keeps track of the state position within a "word" unit; q^{tr} and w^{tr} (both having cardinality 2) represent state and

[1] http://ssli.ee.washington.edu/~bilmes/gmtk/auroraTutorial.tar.gz

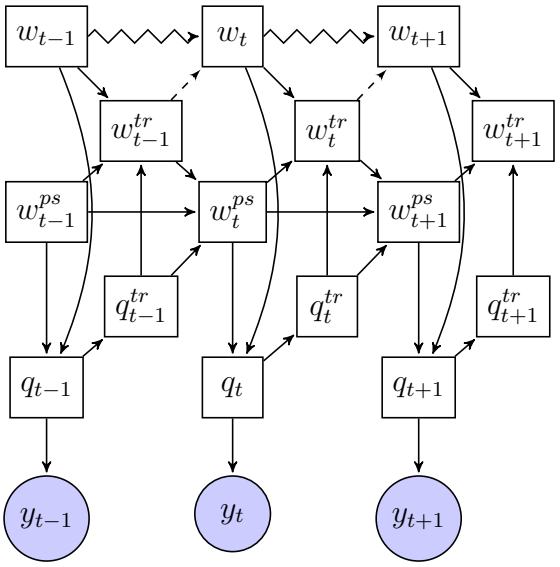


FIGURE 3.2: Architecture of the Dynamic Bayesian Network which is taken as a starting point in this chapter.

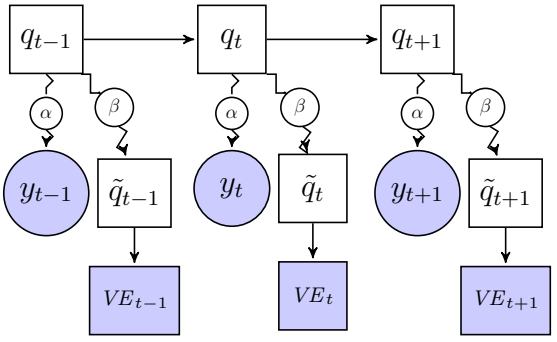


FIGURE 3.3: Observation and state layer of the dual-input DBN.

word transitions, respectively; the short pause consists of a single state which is tied to the middle state of the silence; y denotes the observed MFCC vector; q represents the state-id and has cardinality $S = 11 \times 16 + 3 = 179$.

White symbols in Fig. 3.2 represent hidden variables, while observed variables are shaded; discrete variables are represented by squares and continuous variables by circles. Furthermore, straight lines represent deterministic relations, while zigzagged lines indicate probabilistic relations. Relations between continuous and discrete variables are modeled using GMMs; relations between discrete variables are described in terms of discrete conditional probability tables (CPTs). Dashed lines correspond to a switching parent dependency.

3.2.2.2 Dual-input DBN

In our approach we prefer to combine the likelihoods from the GMMs with the SC state posteriors in the form of virtual evidence (VE). In contrast to the other approaches discussed in the introduction, no additional transformations or dimensionality reduction procedures are required and the external information provided by SC can be used as is. Perhaps more importantly, the VE approach makes it possible to train the GMMs taking into account the beliefs of the external knowledge source, in a way that complies with the Bayesian framework.

As explained in [114] and [115], external, probabilistic evidence about the value of a variable in the network can be incorporated by introducing an observed variable VE, which is a child of the variables for which one has evidence (in our case q_t) and by setting $p(\,VE = 1|q_t) = h(q_t)$, where $h(.)$ is a valid probability density function that represents the available probabilistic evidence. In our case, $h(q_t)$ is set equal to the probability estimates $p(q_t|SC_t)$ obtained from the SC system. To avoid numerical problems during the computation of log-probs (the format required to insert the SC input into the DBN), the (many) zeros in $p(q_t|SC_t)$ are substituted by a floor value of 10^{-30}.

The input stage of the dual-input DBN that we created for combining the MFCC input and the state probability estimates from the SC system is shown in Fig. 3.3. As before, the observed variable y_t denotes the MFCC feature vector at time frame t, and the dependency between q_t and y_t is modeled by GMMs. In parallel to y_t, the SC input is inserted as a second input stream in the form of VE. Furthermore, we introduce weights for both streams, α and β respectively, which allow us to control the impact of either input. The role of these stream weights will be discussed in more detail later. Both the y_t and the VE input are sampled at a rate of 100 observations per second; the two input streams are strictly synchronized.

Due to the fact that the GMM and the SC system use intrinsically different classification procedures, it is unlikely that the estimates of the parallel streams are always in full agreement. To handle possible disagreements, we introduced a hidden node $\tilde{q}_t$. Disagreements between the SC posterior estimates and the state sequence that is optimal in the presence of all other evidence is modeled by a 179×179 CPT (indicated as SC-CPT in the remainder of this chapter). Thus, at the state level, the network will see the VE input in the form of the

product of the frame-vectors provided by the SC system and the SC-CPT, i.e., $p(VE_t|q_t) = \sum_{\tilde{q}_t} p(VE_t|\tilde{q}_t) \cdot p(\tilde{q}_t|q_t)$.

Since the mechanisms underlying the GMM-based and the SC-based classifiers are sufficiently different, we treat the likelihoods obtained from the GMM and the SC streams as if they are conditionally independent. Thus, we assume that the joint likelihood $p(y_t, VE_t|q_t)$ in Fig. 3.3 is equal to the product of the likelihoods of the individual inputs:

$$p(y_t, VE_t|q_t) = p(y_t|q_t)p(VE_t|q_t) \tag{3.2}$$

where $p(y_t|q_t)$ and $p(VE_t|q_t)$ are the contribution of GMM and SC, respectively. For the joint probability in Eq. (3.1) the addition of the VE input boils down to replacing the term $p(y_t|q_t)$ by $p(y_t|q_t)p(VE_t|q_t)$.

The SC system yields state posterior probability estimates, rather than state likelihoods. Since the dual-input DBN requires its inputs in the form of likelihoods, $p(q_t|SC_t)$ must be converted to the likelihood $p(SC_t|q_t)$ through division by the state priors $p(q_t)$ [115]. It appears that in the AURORA-2 task all state priors are virtually identical, so that division by the prior is (nearly) equivalent to scaling the posterior probabilities with a frame dependent factor. Scaling all components in the likelihood vector does not influence the decoding result, since all hypotheses in the search will be penalized or boosted by the same amount, keeping the competition between these hypothesis intact. Therefore, we can treat the posterior probability estimates $p(q_t|SC_t)$ as if these were scaled likelihoods. As a result, the stream merging to be discussed in this chapter takes place in the log-likelihood domain.

After converting the factorized joint likelihood to the log domain, the state sequence $Q^*_{1:T}$ returned by a Viterbi decoding which maximizes the joint log-likelihood LL can be expressed as:

$$\begin{aligned} Q^*_{1:T} &= \underset{q_t}{\text{argmax}}\{LL\} = \\ &= \underset{q_t}{\text{argmax}} \left\{ \alpha \sum_t \log(p(y_t|q_t)) + \beta \sum_t \log(p(VE_t|q_t)) \right. \\ &\qquad \left. + \sum_t \log(p(Rest_t)) \right\} \end{aligned} \tag{3.3}$$

The term $\sum_t \log(p(\text{Rest}_t))$ in Eq. (3.3) summarizes the contribution of all the remaining nodes "above" the state level in the DBN in Fig. 3.2, i.e., the contribution of the state transition probabilities and the language model to the scores of the best path.

The coefficients α and β, the weights assigned to the GMM and SC inputs in Fig. 3.3, make it possible to vary the contributions of the parallel inputs. However, these coefficients have a different status during training and decoding. During training it is essential that all probability distributions in the network represent true probability functions (i.e. sum to unity). Otherwise, stable training results cannot be guaranteed. This requirement can only be met if $\alpha = \beta = 1$ during training with two parallel inputs [109, p. 22]. During decoding, however, we have more freedom, because the Viterbi search that maximizes Eq. (3.3) can yield consistent results for arbitrary values of $\alpha \geq 0$ and $\beta \geq 0$. It is not required to impose $\alpha = \beta = 1$, nor to impose a limitation on the sum $\alpha + \beta$, to guarantee consistency.

From Eq. (3.3) it can be seen that using values $\alpha + \beta \neq 1$ affects the balance between the first two terms on the one hand and the third term on the other. This is reminiscent of the language model factor that is present (and must be optimized) in conventional ASR systems. In the AURORA-2 task not only the prior probabilities of the states are almost constant, but the same holds for the (non-zero) state transition probabilities. Therefore, it seems to be safe to assume that the impact of the third factor in Eq. (3.3) can be ignored in the experiments. Doing this will simplify the design of experiments aimed at finding the optimal values of α and β. We will come back to this issue in the Discussion section.

3.3 Set-up of the Experiments

3.3.1 Database

In our experiments, we use the *multi-condition training set* in the AURORA-2 database [11] for training the GMMs and CPTs in the DBN. This set contains 8440 connected digit utterances from the TIDIGITS database, spoken by 55 male and 55 female speakers. The utterances are artificially corrupted with four noise

types (subway, babble, car, and exhibition hall), with SNRs ranging from clean to SNR = 5 dB.

For testing we used test set 'A' (utterances corrupted by the same noise types as in the multi-condition training set) and test set 'B', containing utterances corrupted by four other noise types (viz. restaurant, street, airport, train station), which are not comprised in the training materials of the GMMs and which are also not covered by the noise dictionary employed in the SC system. Both test set 'A' and 'B' contain 4004 utterances consisting of a sequence of one to seven digits, 1001 utterances for each noise type. All utterances occur in seven noise levels, viz. clean, and SNR = 20, 15, 10, 5, 0, and -5 dB.

3.3.2 Features

The MFCC input to the DBN consisted of 39 dimensional vectors containing 12 cepstral features plus a separate log-energy coefficient, as well as the corresponding first and second order delta coefficients. They were based on a 23 band Mel-frequency spectrum, using a Hamming analysis window of 25 ms and a frame shift of 10 ms. Subsequently, all coefficients were mean and variance normalized for each utterance.

As described in Section 3.2.1, for the SC input we used the likelihood (scaled posterior probability) estimates that were produced by the system described in [69].

3.3.3 DBN Training

Training of the DBN amounts to learning the CPTs (connecting the discrete nodes) and the GMMs (connecting the continuous input y_t to the discrete node q_t in Figs. 3.2 and 3.3) that maximize the likelihood of the training data. Thus, it involves the simultaneous estimation of all functions which describe the probabilistic relations corresponding to the edges in the network. In our experiments we focus on training the "acoustic" models, viz. (1) the GMMs that characterize the relations between y_t and q_t, and (2) the SC-CPT used to map the state probability estimates from the SC system (i.e., $p(VE_t|\tilde{q}_t)$) to q_t.

During training the GMMs, Gaussians were split once the difference of the likelihoods between two iterations did not differ more than 2%; our final GMMs consisted of up to 64 mixture components. For our experiments we used three slightly different DBNs which involved two sets of GMMs and two SC-CPTs.

1. The first set of GMMs was trained without the SC input being present (by setting $\alpha = 1 \ \wedge \ \beta = 0$ during training), which effectively defaults to using the network in Fig. 3.2.

2. The second set of GMMs was trained with the SC input being present $\alpha = 1 \ \wedge \ \beta = 1$, and using the fixed identity matrix as CPT.

3. For training the SC-CPT in the presence of GMMs ($\alpha = 1 \ \wedge \ \beta = 1$), a 179×179 matrix was initialized uniformly and training continued until the likelihood of generating the training data in successive iterations did not differ more than 2%.

The CPTs gathered in the term $\sum_t \log(p(Rest_t))$ in Eq. (3.3) were trained for each of the three DBNs that contained a different combination of the two sets of GMM models and the SC-CPT. This resulted in three (slightly) different sets of CPTs, each of which will be used in the proper context in decoding.

3.3.4 Design of the Experiments

Using the two sets of GMMs and the SC-CPTs we conducted a number of experiments in which we varied the fusion of the two input streams during *decoding*. In the first experiment we aimed to verify the correctness of the DBN implementation and to set a baseline for the performance of a GMM-only and an SC-only decoder (cf. Section 3.4.1). In the second experiment we compared the effect of simply fusing the two baseline systems with the effect of jointly training the GMMs and the SC-CPT. The results of this experiment triggered an in-depth analysis of the distributions of the GMM and VE likelihoods in the two parallel input streams (cf. Section 3.4.2). Next, we carried out a set of experiments in which we varied α and β (cf. Section 3.4.4). In a final set of experiments (cf. Section 3.4.5) we manipulated the SC input vectors in addition to changing the weights α and β during decoding. A more detailed motivation for the latter two sets of experiments will be presented in the corresponding subsections.

3.4 Results

3.4.1 Baselines

To set a baseline and to verify the correctness of the DBN implementation we created two single-input baseline systems. The first baseline system uses the GMMs that were trained without the presence of the SC input. The second baseline system only uses the likelihood estimates from the SC-system as its input, in combination with the identity matrix I for the SC-CPT. The word error rates obtained with these systems, averaged over the four noise types in test set A and B respectively, are shown in the top panel of Table 3.1. The results for the GMM-only system are in the row labeled *G(base)*; the row labeled *S(base)* contains the results of the SC-only system.

The performance of the GMM-only system is comparable to state-of-the-art HMM systems [116]. The performance of the SC-only system is virtually identical to the system in [69]. It can be seen that the *G(base)* system outperforms *S(base)* in almost all conditions (both for test set A and B). The exception is at SNR$= -5\,$dB in test set A, where *S(base)* performs substantially and significantly better than *G(base)*.

3.4.2 Combination of Individually and Jointly Trained Models

Next, we investigated to what extent the two different input streams provide complementary information. We first combined the independently trained baseline systems in a straightforward manner, by creating a dual-input decoder that uses the GMM models that were trained without the presence of the SC-input in combination with the SC-input and the identity matrix for the SC-CPT, and setting $\alpha = \beta = 1$. The word error rates obtained with this system are shown in the row *G/S(indiv)* in the second panel of Table 3.1. The row *G/S(joint)* in that panel shows the results obtained for the dual-input system in which the GMM models *are* trained with the SC-input present, and where the SC-input is used in combination with the trained SC-CPT. Again, during decoding we set $\alpha = \beta = 1$.

TABLE 3.1: Word error rates (in %) as obtained with the DBN system. Top panel: single-input systems. G(base) is the GMM-only system, S(base) the SC-only system. Second panel: dual-input systems. G/S(indiv) uses the GMMs and SC-CPT copied from the *single* input systems; G/S(joint) uses the jointly trained GMMs and SC-CPT. Differences between G/S(indiv) and G/S(joint) which are statistically significant at the $p = 0.95$ level are marked with an asterisk. Third panel: G/S(sel_w) word error rates obtained with a manually selected pair of weights $(\alpha, \beta) = (0.4, 0.2)$ (see Section 3.4.4). Fourth panel: To allow a coarse assessment of the statistically significant differences the bottom row shows the 95% confidence intervals based on the WERs of G(base).

			SNR(dB)↘	test set A						test set B					
α	β	system	clean	20	15	10	5	0	-5	20	15	10	5	0	-5
1	0	G(base)	0.63	0.77	1.19	2.48	6.20	20.38	52.89	0.87	1.52	3.28	8.58	26.32	61.78
0	1	S(base)	7.92	8.91	9.55	10.84	13.75	23.21	43.95	8.92	9.65	12.24	19.45	35.55	63.88
1	1	G/S(indiv)	0.54	0.78	1.08	2.34	5.69	17.23	46.35	0.84	1.45	2.97	8.67	25.19	58.37
1	1	G/S(joint)	0.55	0.77	1.08	2.24	5.53	16.93	45.14*	0.79	1.21*	2.93	7.92*	23.76*	56.92*
0.4	0.2	G/S(sel_w)	0.48	0.84	1.17	2.34	5.24	15.29	42.03	0.72	1.20	2.62	7.04	21.46	53.80
95% conf. interval			0.14	0.15	0.19	0.27	0.41	0.69	0.86	0.16	0.21	0.31	0.48	0.76	0.83

From Table 3.1 it can be seen that the straightforward combination of the two input streams (*G/S(indiv)*) improves most results over the individual systems. Joint training has a small, additional advantage over the straightforward combination of the two input streams. Thus, it appears that the two streams do contain complementary information and that the system is able to learn the systematic differences between the state assignments $\tilde{q}$ of the SC system and the eventual state assignments q. A closer inspection of the trained SC-CPT indeed showed that it deviates somewhat from the identity matrix. The main observation is that the state alignments of the SC system differ from the reference forced alignment: the SC system assigns an appreciable part of the probability mass to both neighboring states, and sometimes one of the neighbors obtains the highest posterior. To compensate for the different alignments, the trained SC-CPT is a narrow (roughly 3 states wide) banded matrix, rather than a true diagonal identity matrix. No other systematic discrepancies were observed.

Another interesting observation that can be made from Table 3.1 is that combining GMM and SC reduces the 0.63% error rate of *G(base)* in clean speech to 0.55%. Despite the fact that the error rate of the *S(base)* system in the clean condition is as high as 7.92%, the SC stream apparently can compensate for some of the errors that results from the GMM-only system. However, there is also one condition, i.e., SNR=-5 dB in test set A, where even the best dual-input system performs worse than *S(base)*. Although the difference in WER of 45.14% vs. 43.95% only approaches statistical significance, this finding still calls for an explanation.

3.4.3 Asymmetric Effects of GMMs and SC

To explain the asymmetric effects of the fusion of the two input streams (*S(base)* helping *G(base)* in the clean conditions, and *G(base)* hindering *S(base)* in the most noisy condition) we need to consider the distributions of the likelihoods $p(y_t|q_t)$ and $p(\,VE_t|q_t)$. From Eq. (3.3) it can be inferred that the relative impact of $p(y_t|q_t)$ and $p(\,VE_t|q_t)$ is determined by the average shape of these likelihood vectors: if one of the distributions tends to divide the total probability mass over a large number of states, while the other concentrates the probability mass in a small number of states, the latter is likely to have a much stronger impact than the former.

To investigate commonalities and differences between the distributions obtained from the GMMs and the SC system, we computed the statistics of the maximum

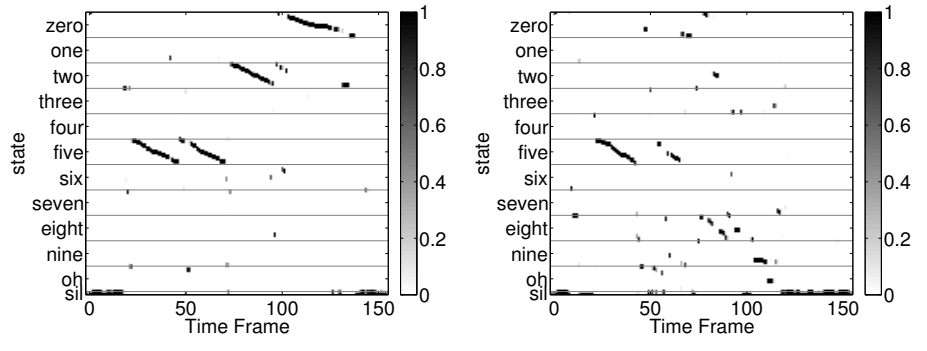
TABLE 3.2: *The average values of the maxima in the likelihoods $p(y_t|q_t)$ and $p(\,VE_t|q_t)$ obtained for four SNR conditions in test set A. GMM(indiv): using the GMMs trained in the absence of SC; SC(indiv): using the identity matrix as SC-CPT; GMM(joint) using the GMMs trained in the presence of SC; SC(joint): using the trained SC-CPT.*

SNR(dB)	clean	10	0	-5
GMM(indiv)	93.47%	92.21%	91.43%	90.96%
SC(indiv)	25.02%	20.28%	17.00%	14.59%
GMM(joint)	92.51%	90.39%	89.28%	88.48%
SC(joint)	28.30%	23.58%	19.77%	17.49%

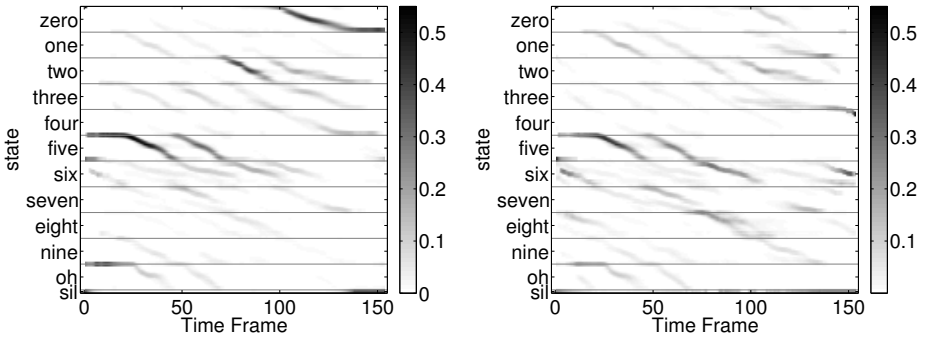
values in $p(y_t|q_t)$ and $p(\,VE_t|q_t)$ for the utterances in test set A in four SNRs, viz. clean, and SNR=10, 0 and -5 dB. For each speech frame we computed the likelihood $p(y_t|q_t)$ by computing the likelihoods of the observed MFCC vector for each of the 179 trained GMMs and normalizing them in such a way that they sum up to one. The VE likelihood $p(\,VE_t|q_t)$ was obtained by multiplying the observed state likelihood vectors $p(\,VE_t|\tilde{q}_t)$ from the external SC-system by the DBN-internal SC-CPT $p(\tilde{q}_t|q_t)$. Subsequently, we computed the statistics of the maximum values in $p(y_t|q_t)$ with both sets of GMM models, and the maxima in $p(\,VE_t|q_t)$ with the identity matrix I and the trained SC-CPT. If the value of the maximum averaged over a complete test set is large, most of the probability mass is concentrated in a single state q_t, which corresponds to a very sharp distribution.

The results are shown in Table 3.2. The most striking observation from this table is that in $p(y_t|q_t)$ most of the probability mass gets assigned to one state (out of 179). Although slightly less, this bias towards a single state also exists with the GMMs trained in the presence of the SC input. Moreover, it can be observed that the GMMs retain this tendency even in noisy speech. In the distributions of $p(\,VE_t|q_t)$ the probability mass is spread over multiple states. Especially in low SNR-conditions the most probable state accounts for less than 20% of the probability mass.

The effect of the different shapes of the distributions of $p(y_t|q_t)$ and $p(\,VE_t|q_t)$ is also illustrated in Fig 3.4, which shows the probability estimates in one utterance (*five-five-two-zero*) in clean and SNR=-5 dB subway noise. The horizontal axis represents time, while the vertical axis represents the states; between each pair of horizontal grid lines there are 16 states for each digit. In the $p(y_t|q_t)$ estimates



(A) GMM, clean (B) GMM, SNR=-5dB



(C) SC, clean (D) SC, SNR=-5dB

FIGURE 3.4: *Normalized $p(y_t|q_t)$ and $p(VE_t|q_t)$ likelihoods from the GMM and SC streams, respectively, for a single utterance 'five-five-two-zero' at clean and SNR=-5 dB.*

(top), there is usually only one candidate state for each time frame, in both the clean and SNR=-5 dB condition. In the $p(VE_t|q_t)$ estimates (bottom), several neighboring states, as well as states from other digits (which presumably show some resemblance with the exemplars of the current digit) receive substantial probability values.

To better understand the findings in Fig 3.4 and Table 3.2, we computed the per-frame entropy of the likelihoods. The results confirmed that, on average, the SC-based VE likelihoods form a flatter distribution than the GMM-based state-likelihoods. Therefore, the finding that at high SNRs the inferior SC input can help the superior GMM input, and that at SNR=-5 dB the inferior GMM input hurts the superior SC input, is probably due to the difference in impact between the terms $\log(p(y_t|q_t))$ and $\log(p(VE_t|q_t))$ in Eq. (3.3) when $\alpha = \beta = 1$.

3.4.4 Balancing the Weights of the Two Input Streams

Since it appeared that the two input streams may have different impacts, we want to investigate whether the performance of the dual-input system can be improved by optimizing the weights of the two input streams during decoding. This can be done by varying α and β. In the linear (probability) domain, α and β act as exponents that affect the shape of the distributions. If $p(\cdot)$ denotes a probability distribution, then $p^{\alpha}(\cdot)$ has a flatter shape for values of $\alpha < 1$, while the shape becomes sharper for $\alpha > 1$. In the log-domain, α and β serve as multiplication factors that affect the dynamic range of the log-prob scores of the two streams and thereby their relative impact on the eventual recognition result.

Using a grid search in which we explored 165 combinations of α and β, we evaluated the word accuracy for both test set A and B at four SNR conditions: clean, 10 dB, 0 dB and -5 dB. In this grid search 11 values for α were chosen in the interval $[0, 1]$, using a step size of 0.1. For β we used 15 values divided into two subranges. The first subrange for β was $[1, 5]$, using a step size 1; this corresponds to increasing the relative difference between 'large' and 'small' probability estimates in the SC output, which should counter the tendency of the GMM stream to concentrate most of the probability mass in a single state. Additionally, we explored β values in the subrange $[0, 0.9]$, with a step size 0.1, to keep correspondence with the values of α used for the GMM-stream. It should be noted that a setting $(\alpha < 1) \wedge (\beta < 1)$ effectively decreases the contribution of the first two terms in Eq. (3.3) relative to the third term, while $(\alpha > 1) \wedge (\beta > 1)$ has the opposite effect.

Fig. 3.5 shows the word error rates as a function of α (along the vertical axis) and β (along the horizontal axis) in the form of filled contour plots. The (red) dots indicate the (α,β)-combinations for which the word error rates were in the bottom 5-percentile of the values obtained across the 165 grid points; this bottom 5-percentile word error rate is shown above each subplot. Successive contour lines represent the distance from these 5-percentile best performance levels: Going from white to black the red contour levels correspond to the p=84%, 95%, 97.5%, 99%, 99.9% and 99.999% (one-sided) confidence intervals. In the black areas the white contour lines show increasing WERs (increase of 1% absolute per contour).

From Fig. 3.5 we can make a number of observations. First, stream weighting is not only important for a dual input system. In a single-input GMM system, represented by the accuracies on the $\beta = 0$ axis, the curvature of the white contour

lines indicates that the best performance for the GMM-only system is typically obtained for a value $\alpha \approx 0.4$. At the same time, it is clear that this flattening of the GMM pdf's does not yield a performance that can compete with the dual input system. The additional SC stream ($\beta \neq 0$) helps to improve recognition results in all {test set, SNR}-conditions. For $\beta = 0$ the WERs are significantly ($p \leq 0.05$) higher in all conditions, except for a small range of $\alpha = [0.2, 0.3]$ in the clean condition.

From the largely horizontal patterns in all sub-figures it can be inferred that in all {test set, SNR}-conditions the performance of the dual-input system is far more sensitive to α than to β. In the two cleanest conditions, there is a wide range of β values within which the performance does not vary significantly once a proper value for α is chosen. In the more noisy conditions the range of β values within which performance does not change significantly is (slightly) more restricted. In addition, it can be seen that, especially in the cleaner conditions, larger values of β make the system less sensitive to the value of α.

The bottom panel of Table 3.1 shows the performance obtained at the grid point (α, β)=(0.4,0.2). This grid point was chosen based on the low average WER level on test set A. It can be seen that only between 10 dB and 20 dB SNR for test set A the performance is slightly worse than *G/S(joint)*; in all other {test set, SNR}-conditions the performance is better than *G/S(joint)*, with (α, β)=(1.0,1.0). Most importantly, in test set A the performance in the SNR=-5 dB-condition now exceeds the performance of *S(base)*; the difference is significant at the $p \leq 0.05$ level. This shows that the performance of the dual-input system can be improved by proper weighting of the contributions of the two input streams during decoding. For test set B, the chosen weight combination also gives better performance than *G/S(joint)* with (α, β)=(1.0,1.0) for all SNRs. This suggests that the beneficial effect of a proper stream weighting generalizes to noise types that were not seen during training.

A comparison of the various SNR conditions in Fig. 3.5 suggests that the value of α must be reduced as the SNR level decreases. Also, it appears that most of the close-to-optimal results are obtained with values $\beta < 1$. If both α and β are smaller than one, the relative weight of the information encapsulated in the $p(Rest)$ term in Eq. (3.3) becomes more important. The 10 dB SNR condition in test set A might seem to be an exception, because here the best results are obtained with $\alpha = 1$. This can be explained by noting that this condition gives the best average

match between the test data and the multi-condition training data. The finding that the best performance on clean speech of the GMM-only system ($\beta = 0$) is obtained with with values $\alpha < 1$ is due to a mismatch between the clean test data and the multi-condition training data.

In the (α, β)-region explored in the grid search, there is an interaction between α and β: the optimal β values vary with α. This relation is not monotonic: Fig. 3.5 shows that in five out of seven conditions close-to-optimal results can also be obtained with values of $\beta > 1$. In the SNR=10, 0 and -5 dB conditions of test set B (of which the noises have been seen neither by the GMMs nor by the SC system), the grid points where the top 5% performances are obtained are located in two disjoint regions, viz. $\alpha \approx 0.3$ and $\beta \approx 0.2$ on the one hand, and $\alpha \approx 0.4 - 0.5$ and $\beta \approx 4 - 5$, on the other. In the former area, the information of the GMM stream dominates the decisions on the most likely state sequence, while the information of the SC stream has more impact in the latter area. This corroborates the conclusion that the two streams do carry different evidence. It also suggests that it is not possible to find a unique set of weights that is optimal for all SNR conditions.

3.4.5 Reducing the Support of SC

Values of $\beta > 1$ emphasize the states that are considered most likely by the SC-system and de-emphasizes the less likely ones. At very high values of β the de-emphasis becomes equivalent to discarding the lowest state probabilities in the SC vector. In previous research on using DBNs for combining SC and GMMs in speech recognition, we retained only a limited number of non-zero SC-coefficients by successively removing the smallest coefficients and subsequently renormalizing the remaining coefficients to sum to one [84, 108]. This procedure was dubbed "reducing the support of the SC vectors". The results suggested that truncating the SC-vector did improve the word error rate in a system with uniform weights for the two streams.

The results in Section 3.4.4 showed that values $\beta > 1$, which also emphasizes the largest coefficients in the SC vectors, improve WERs in the lower SNR conditions. Therefore, we analyze the relation between the two mechanisms for reshaping the SC vectors in more detail. We will refer to the number of non-zero coefficients that are retained as SC-Dim. We investigated the interaction between α and SC-Dim, while keeping $\beta = 1$. We varied α in the interval $[0.1, 1]$, using a step size 0.1. For

TABLE 3.3: Word error rates (in %) of the DBN system for various values of SC-Dim. Top panel: word error rates obtained with the manually selected grid point $(\alpha, \beta) = (0.4, 0.2)$ (see Section 3.4.4). Second panel: word error rates for various values of SC-Dim with $(\alpha, \beta) = (0.4, 0.2)$. (See Section 3.4.5.) Values that are statistically significantly different from the top row are marked with an asterisk. Third panel: 95% confidence intervals [based on G/S(179dim)]

		SNR(dB)↘	test set A						test set B					
SC-dim	system	clean	20	15	10	5	0	-5	20	15	10	5	0	-5
179	G/S(179dim)	**0.48**	0.84	1.17	**2.34**	5.24	15.29	42.03	**0.72**	**1.20**	**2.62**	7.04	21.46	53.80
100	G/S(100dim)	**0.48**	**0.83**	**1.14**	2.37	5.11	**15.12**	41.64	**0.72**	1.22	2.66	6.91	21.13	53.19
50	G/S(50dim)	0.50	0.85	**1.14**	2.35	5.08	**15.12**	41.39	0.75	1.23	2.64	**6.89**	20.99	52.69*
20	G/S(20dim)	0.50	0.85	1.17	2.38	**5.04**	15.19	41.51	0.76	1.23	2.66	6.97	20.95	52.57*
5	G/S(5dim)	0.51	0.90	1.36	2.55	5.26	15.23	41.06*	0.86	1.30	2.80	7.20	**20.73**	**52.31***
2	G/S(2dim)	0.67	0.93	1.56	2.77	5.56	15.39	**40.42***	0.98	1.48	2.91	7.16	20.75	52.43*
1	G/S(1dim)	0.60	0.95	1.47	2.80	5.37	15.78	40.90*	1.02	1.41	2.84	7.05	20.84	52.77*
95% conf. interval		0.11	0.15	0.18	0.25	0.37	0.60	0.82	0.14	0.18	0.26	0.42	0.68	0.83

SC-Dim we selected eight values: $\{1, 2, 5, 10, 20, 50, 100, 179\}$. For each value of SC-Dim, we replaced the vector elements with a rank $\geq$ SC-Dim by the floor value 10^{-30}, after which the vectors were renormalized.

Fig. 3.6 shows the recognition performance as a function of α (on the vertical axis) and SC-Dim (on the horizontal axis) by means of filled contour plots. The sub-plots show the word error rates for test set A and B in four different SNR conditions. Again, starting from the white area, the contour levels correspond to the p=84%, 95%, 97.5%, 99%, 99.9% and 99.999% (one-sided) confidence intervals relative to the performance at the 5-percentile point. The white contour lines in the black areas correspond to ever larger WERs (increase of 1% absolute per contour line).

The first observation that can be made from Fig. 3.6 is the striking similarity between the left hand side of the sub-plots, where SC-Dim > 50, and the right hand side of the plots in Fig. 3.5, where $\beta > 1$. This indicates that the net effects of reducing the support of SC and increasing the influence of the SC stream by the stream weight β are very similar. As a result, we can draw many of the same conclusions as in Section 3.4.4. For example, from the slopes of the contour lines in the black areas in the sub-plots for clean and SNR=10 dB it can be seen that emphasizing the largest entries in the SC vector requires a higher value of α, which corresponds to a higher weight of the GMM estimates relative to the SC estimates.

The fairly sharp transition to higher WERs for SC-Dim < 50 suggests that the 50 coefficients with the lowest rank/highest value all contain some relevant information, except perhaps in the -5dB condition in test set A, where an alternative optimum is present at very low values of SC-Dim. This is the single condition in which the *S(base)* system clearly outperforms the *G(base)* system. Setting entries in the SC vector to the floor value 10^{-30} makes paths through the corresponding states in the Viterbi search (cf. Eq. (3.2)) very costly. Apparently, the few states that are still licensed when SC-Dim is very small are often on the correct path. But it is also clear that the GMMs still contribute useful information for choosing between the small number of candidates that are left.

In contrast to the fairly abrupt changes in WER −when decreasing SC-dim from 179 down to 1− that are evident from Fig. 3.6, in [84, 108] we found that the WERs changed relatively gradually. Typically, the lowest WER were found for for values of SC-Dim at the lower end of the 1-179 range. This seeming discrepancy can be explained by noting that we always used $\alpha = 1$ in our previous studies. This

corresponds to the top horizontal line in Fig. 3.6. Along this line the differences in WER are less obvious and occur much more smoothly than for lower values of α. As in the previous studies, we see that with decreasing SNR the lowest WERs are found for lower values of SC-Dim. The detailed analysis in this chapter shows that it is essential to have a complete picture, that uncovers the impact of the statistical properties of the streams that are combined. An analysis that is limited to part of the space spanned by the parameters investigated in this chapter, which ignores the potential effects of different distributions of SC and GMM likelihoods, can give rise to misinterpretations.

In Section 3.4.4 it was found that the best results were obtained with values $\beta < 1$, which de-emphasize the largest coefficients in the SC vector. Therefore, we investigated the effect of using SC-Dim < 179 in combination with stream weights that differ from one. In Table 3.3 we show the WER results obtained with the previously selected 'optimal' values $(\alpha, \beta) = (0.4, 0.2)$, for several values of SC-Dim. The WERs indicate that the results obtained with SC-Dim > 20 or maybe even > 5 do not differ significantly from those obtained with SC-Dim $= 179$. The fact that in the lowest SNR conditions the best results are obtained with very small values of SC-Dim is in accordance with the previous finding that in these conditions close to optimal results can be obtained with values $\beta \approx 5$. Taken together, these results suggests that the contributions of the coefficients in the SC vectors with ranks between 20 and 50 are marginal, and that most of the time the coefficients with ranks ≤ 5 indicate the correct state. Because the SC state likelihoods are computed from a *sparse* combination of exemplars, this does not come as a surprise.

3.5 Discussion and Conclusions

In this chapter we tried to develop a principled explanation for the results of previous experiments in which we observed improvements in WER for the AURORA-2 tasks by means of several ways of combining likelihood scores obtained from a GMM with independently obtained state likelihoods in the form of Virtual Evidence in a DBN. The reason for wanting to combine GMMs and SC is that GMMs perform very well in the high SNR conditions, while SC shows superior performance in the low SNR conditions. One of the reasons for using the VE option in a DBN is that this makes it possible to simultaneously train GMMs and a CPT that inserts SC estimates into the DBN. Since unexpected results may

occur when the conditional probability functions on the edges of the DBN are not proper probability distributions, it is imperative to keep the weights of the parallel inputs equal to one during training. However, in decoding the stream weights can be optimized with the only constraint that the weights must be ≥ 0. Thus, a large part of the research focused on optimizing these weights during decoding.

From our results in Table 3.1 it can be seen that the dual-input system with weights equal to one for both inputs outperformed both individual systems in all SNR conditions in test sets A and B, except in the SNR=-5 dB (worst) condition in test set A. It can also be seen that joint training of the GMMs and the CPTs in the dual input DBN only resulted in a marginal improvement over fusing the individually trained systems. Importantly, joint training did not remove the inferior performance of the dual-input system in the -5 dB condition in test set A. Somewhat surprisingly, a small (although not statistically significant) performance gain was observed for the dual-input system in the clean condition, in which the GMM-system outperformed the SC-system by a wide margin. Apparently, the SC-stream can occasionally help the GMM stream, even if it (as an individual stream) leads to inferior performance.

The asymmetric behavior of the fusing of the GMM and SC systems in the clean and -5 dB SNR condition in test set A gave rise to an in-depth analysis of the statistical properties of the GMM state likelihood scores and the SC state posterior probability stimulates. It appeared that these properties are very different: while the GMM estimates tend to concentrate the lion's share of the total likelihood in a single state, the SC estimates always attribute similar probabilities to several acoustically similar states. These intrinsically different properties make it necessary to assign different weights to the two streams so that they can make optimal contributions during decoding. By optimizing these weights it is possible to construct a dual-input system that outperforms the best individual system in all conditions, including the -5 dB SNR condition in test set A. However, we did not succeed in finding a unique set of weights that provide optimal results in all conditions. We also found that it can be advantageous to 'flatten' the vector of state likelihoods obtained from the GMMs, which – everything else being equal – corresponds to increasing the weight of the non-acoustical part of the DBN during decoding.

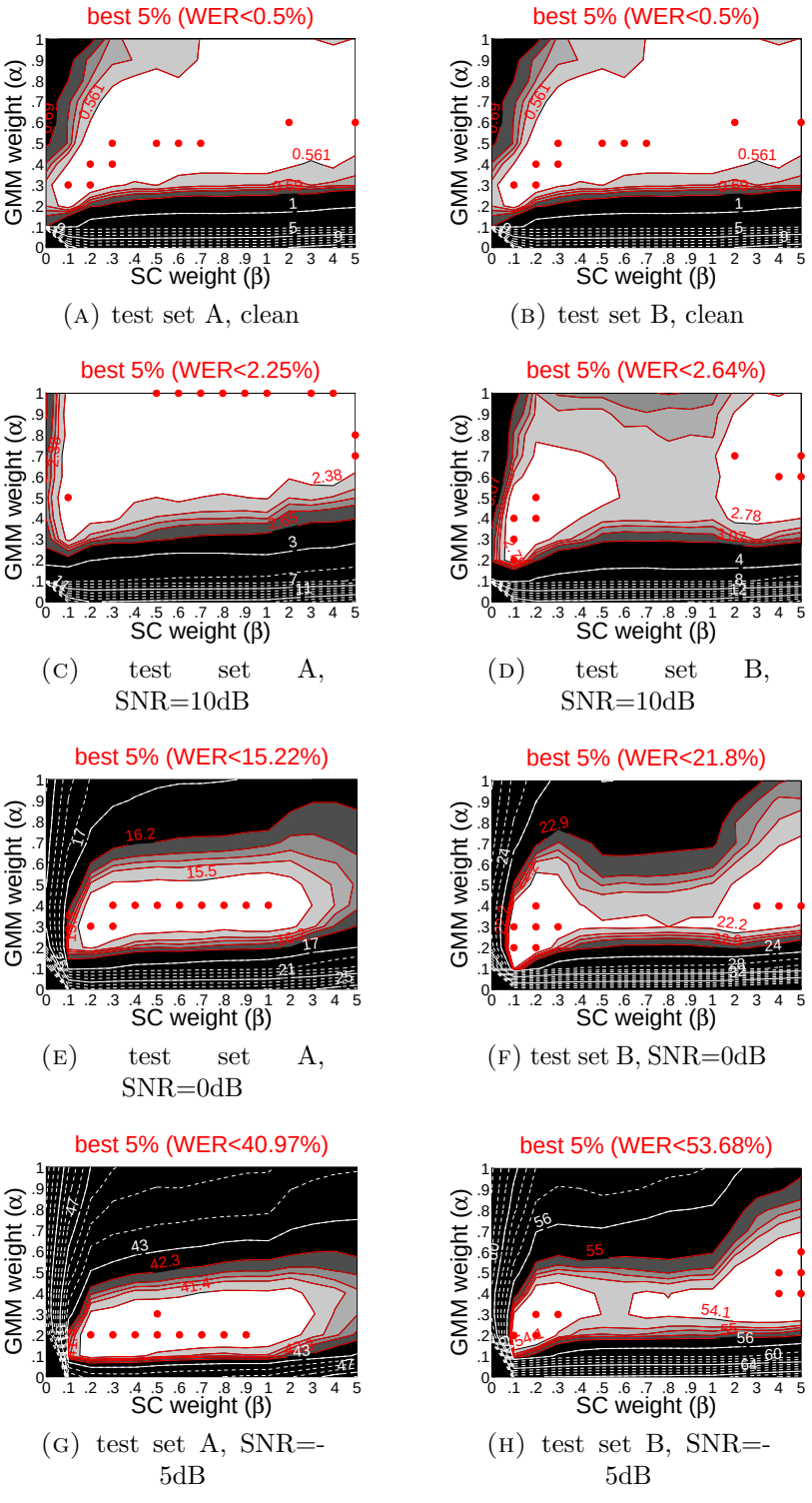
In previous experiments with fusing parallel input streams at the state probability level using state estimates obtained from MLPs or GMMs there was no need for optimizing stream weights during decoding [100–104]. Therefore, it might be

argued that the need for optimizing the weights is an unfortunate side effect of the way in which the SC system computes state probability estimates and that the results in this chapter do not generalize. However, the emergence of novel classifiers in the sparse representation framework and in other machine learning frameworks is likely to introduce additional systems that show promising performance in adverse conditions, while still producing probability vectors with an entropy that is much higher than what is usually obtained from MLP and GMM systems. The need for fusing systems with widely different average entropy outputs is also present in other application domains. We believe that the results presented in this chapter can help guide future efforts in fusing such differing systems.

As explained in Section 3.2.2.2, joint training with the input streams can only be guaranteed to yield consistent results with stream weights $\alpha = \beta = 1$. However, the finding that reducing the support of the SC stream is tantamount to using a value $\beta > 1$ in decoding opens the possibility for also changing the weights of the streams during joint training, without jeopardizing the stability of the results. After re-normalization a truncated SC input is still a valid VE input. Joint training in a condition in which the statistical properties of the two input streams are more similar might well be more effective in identifying the useful information in the joint streams. Therefore, it would be premature to conclude that joint training is of little added value.

Several other avenues for future research exist. Obviously, integrating the recent improvements of the SC approach reported in Gemmeke et al. [117], Gemmeke and Van hamme [118], Hurmalainen et al. [119] are expected to improve the performance of the dual-input recognizer. A more fundamental line of research focusses on the effective combination of streams. As we already discussed in Section 3.4.4, choosing $\alpha < 1 \wedge \beta < 1$ amounts to reducing the acoustic evidence relative to the information encapsulated in the model topology. The topology used in our experiments makes it possible to assign different weights to the streams, and the combined effect of these weights determines the relative impact of the acoustic and non-acoustic probability distributions in the network. In future experiments it might be advantageous to introduce a separate control mechanism for adjusting the relative input stream weights on the one hand, and the weights assigned to acoustic evidence relative to the weight of the word model topology and language model on the other.

The experiments described in this chapter have shown that it is not possible to find a unique set of parameters that yields superior results in all SNR conditions in both test set A and B. This strongly suggests that it is necessary to develop adaptive procedures that can find locally optimal values of the parameters. This, too, will be a topic of future research. Instead of static weighting, different dynamic weighting schemes will be studied in the next chapter to cope with a large SNR range.



(A) test set A, clean

(B) test set B, clean

(C) test set A, SNR=10dB

(D) test set B, SNR=10dB

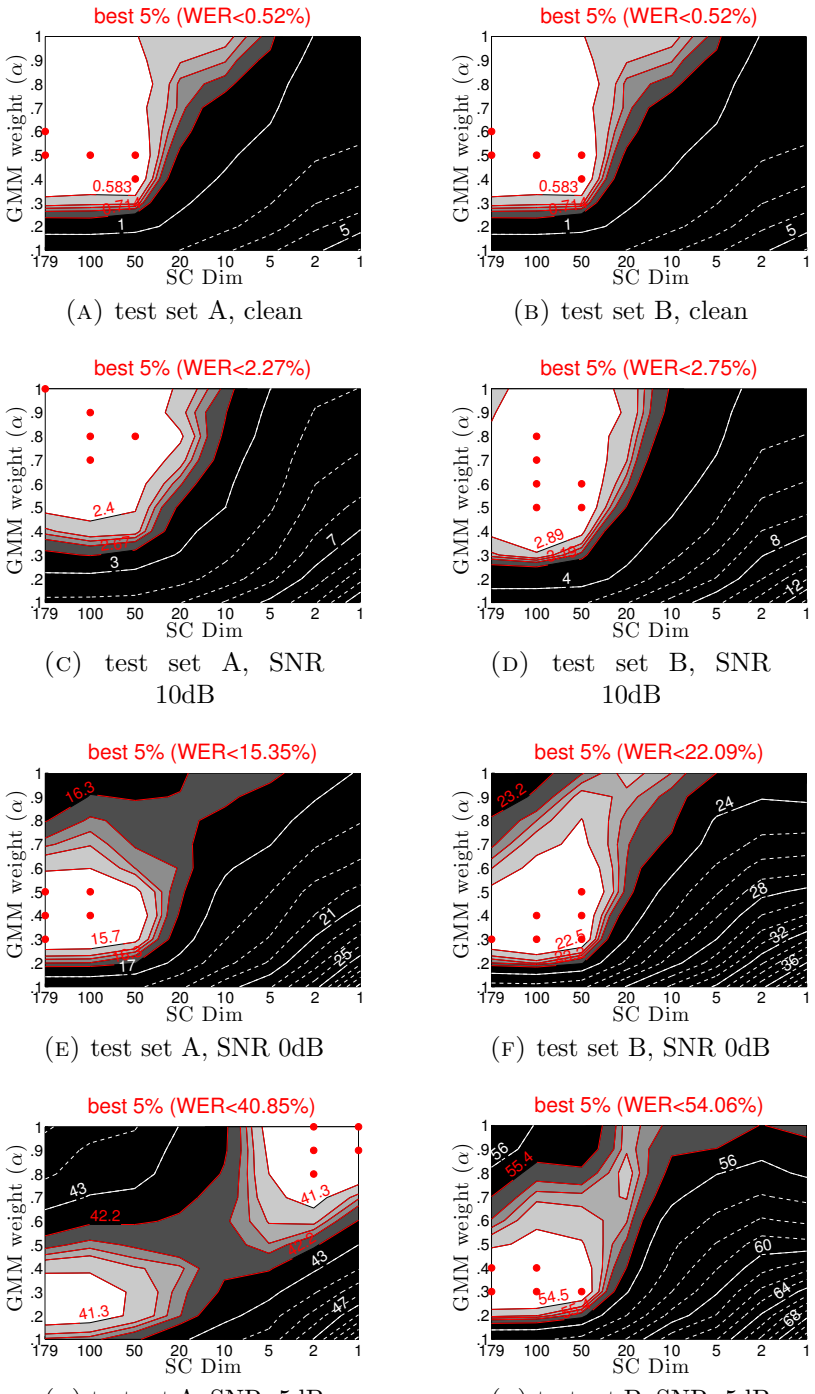
(E) test set A, SNR=0dB

(F) test set B, SNR=0dB

(G) test set A, SNR=-5dB

(H) test set B, SNR=-5dB

FIGURE 3.5: *Ranges of α and β for which statistically similar word error rates are obtained (using the full dimensional SC input). Different contour lines represent distances from the top 5-percentile performance levels in terms of confidence intervals (see text). The (red) dots represent (α, β)settings which result in word error rates below the bottom 5-percentile level (this level is denoted above each subplot). For a further explanation see the text.*



(A) test set A, clean

(B) test set B, clean

(C) test set A, SNR 10dB

(D) test set B, SNR 10dB

(E) test set A, SNR 0dB

(F) test set B, SNR 0dB

(G) test set A, SNR -5dB

(H) test set B, SNR -5dB

FIGURE 3.6: *Word error rates obtained with a grid search decoding across α and SC Dim.*

Chapter 4

Multi-stream System Combination with Confidence-based Adaptive Weights for Robust ASR

4.1 Introduction

Designing novel ASR systems that can outperform state-of-the-art systems in both clean and noisy conditions is a challenging task. For example, the results in [69] show that a Sparse Classification (SC)-based ASR system operating on Mel-band energy spectra can outperform a traditional MFCC-based Gaussian Mixture Model (GMM)-based ASR system in very noisy conditions; in clean conditions, however, the conventional GMM system outperforms the SC system. In order to obtain a recognition system that shows optimal performance in both conditions, it seems attractive to exploit the complementary information that is provided by different types of acoustic features and/or classification strategies that provide state posterior probabilities, sequences of which serve as input to the word search module.

Obviously, combining features and/or classifier outputs is not a novel idea. In the past, several approaches have been proposed that perform fusion at the level of features [120–123], at the level of probabilities [102–104, 124, 125], or at the level of hypotheses [126–129]. In this chapter we started with fusions at the probability level and we use the probability estimates that result after fusion as input for a conventional Viterbi decoder. The eventual model is evaluated using the Word

Error Rate (WER). In addition, we investigate several issues that up to now have escaped attention in the speech technology literature. Specifically, we focus on a number of intricacies that may be encountered when fusing streams of which the probability estimates exhibit very different statistical distributions.

When fusing multiple streams at the probability level, ideally, one would want to be able to obtain the *joint* probability distribution of the outputs of all classifiers directly. In the case of state posterior estimates in a speech recognition system that has to operate in a multitude of noisy environments, this is virtually impossible. Therefore it is common practice to combine the outputs of individual classifiers, using functional combinations valid under certain assumptions concerning stream independence. In the past, various techniques for fusing probability estimates have been proposed, including the SUM, PRODUCT, MAX and MIN rules ([130–133]). All these techniques make assumptions about the data to be merged.

Applying the PRODUCT rule to merge probabilities is tantamount to assuming that the classifier outputs are conditionally independent, and that none of the classifiers makes gross errors. However, especially in a situation where classifiers must generalize to unseen conditions, there is a non-negligible risk that gross errors do happen. When the PRODUCT rule is used, estimates close to zero from one classifier will effectively cancel out all estimates from other classifiers. Instead of producing a fusion result with low confidence since the classifiers disagree, the result is just dominated by the classifier that produced the close-to-zero estimate. For this reason, particularly in situations where the estimates of one or more classifiers are likely to be error prone, the SUM (actually: average) rule might be preferable [132, 133].

One of the main findings reported in [132] was that back-end decoders used for further processing the probability estimates such as a Viterbi decoder are often not very sensitive to the approximation errors that result from the SUM rule. In this chapter, we focus on PRODUCT and SUM rules from which most of the other combination rules are derived [132]. We will investigate to what extent the PRODUCT and SUM rules can be used for harnessing the complementary information in the state posterior estimates of (1) a Multi-layer Perceptron (MLP) classifier, and (2) an SC classifier. For each time frame we compute a weighted combination of the posterior probability estimates produced by the two systems. In addition, we explore to what extent local properties of the posterior estimates can be exploited to automatically switch between the two rules.

The search for the optimal weighting of classifier outputs is an issue that has previously been discussed in the literature (e.g. [50]) and should primarily be considered as a way to account for the fact that classifiers may differ in their "trustworthiness". It is evident that the trustworthiness of a classifier may differ due to the use of different features or due to the use of intrinsically different classification principles when generating the streams. Particularly the latter circumstance must be expected to result in probability estimates with different statistical distributions.

In most previous research on classifier fusion at the probability level, the used classifiers employed the same classification principle, the major differences being the input features on which the classifiers operate. For example, [134] and [135] combined MLP classifiers with different acoustic features. Under this condition it is safe to assume that the relative trustworthiness of the probability vectors at the output of these classifiers can be measured by means of the same confidence measure. It has been shown that, when using MLP classifiers that are trained to assign the bulk of the probability mass to a unique state, a frame-wise, between-stream comparison of the inverse or minimum *entropy* of the probability vectors provides a useful means for estimating trustworthiness [50]. Here, however, we address a more complex situation. Here we combine the probability estimates produced by two different classifiers based on very different classification strategies: an MLP and an SC classifier [69], since we expect these classifiers to provide more complementary information than very similar classifiers would do. Contrary to the MLP, the SC system is *not* trained to concentrate the probability mass in a single state. In fact, our previous research on the AURORA-2 task [11] has shown that the SC system tends to divide the probability mass relatively evenly over a number of states that are acoustically similar [107, 136]. Moreover, the state probability distributions for the SC and the MLP classifier have been shown to have a very different shape [108], which means that inverse entropy may not be a good way for estimating the trustworthiness of the posterior estimates of MLP and SC classifier at the same time. Therefore, we will propose a different method for estimating the relative trustworthiness of the MLP and SC estimates.

The trustworthiness of a stream is not only classifier dependent, but also dependent on the data it is confronted with. If the trustworthiness of the streams does not change substantially across training, development and test data, it is probably adequate to derive static (i.e. frame-invariant) weights from held-out training data. However, if the trustworthiness does change substantially, either within or between

the conditions in which different subsets of data were recorded, static weights will no longer be optimal and some form of dynamic weighting should be considered.

In the case of dynamic weighting there are basically two options. One may specifically estimate weights for each relevant test condition, in combination with a procedure for determining (at testing time) the condition in which new test data are recorded. Alternatively, one may develop a weighting scheme in which the weights are determined on the basis of some local property of the input data during testing [50]. The latter avoids the need for error-prone attempts to determine the operating condition. In this chapter, we will compare the merits of different local weighting methods (using some frame dependent, statistical property) with a global, signal-to-noise ratio (SNR) based weighting approach.

In this chapter, we try to shed some light on the various aspects of the stream combination problem by a set of experiments. Considering the differences in both performances and probability distribution of SC and MLP, a straightforward merge will hardly be optimal. Therefore, given the fact that we know beforehand that the stream which gives the best posterior state probability estimates differs dependent on SNR, our first aim is to know to what extent the optimal stream weights depend on SNR. This benchmark experiment will serve to provide upper limit for the word accuracy by using oracle knowledge about the SNR. Secondly, owing to the big statistical difference of two streams, using only one traditional time-variant weighting scheme, such as inverse entropy-based approach, may not work. We introduced a novel data-driven approach as an alternative dynamic weighting way to adjust weights for each individual stream independently in different SNR conditions, pursuing to the oracle weights and accuracies at each SNR in the first experiment. Additionally, besides the time-variance weighting scheme, we further investigated whether the combination rules can also be selected in a dynamic fashion. Given the fact that the SUM and PRODUCT RULES ARE more conceptually legitimate at low and high SNRs respectively, our last task is to explore an algorithm which can pick the more suitable combination rules automatically on the fly, aiming to quantitatively measure in which condition either of the combination rules should be adopted. In short, a hybrid system will be developed with time-variant weights and combination rules. The dynamic solution would be based on local information (posteriors at current frame) and it should be resistant with the diversity of the input streams.

The rest of the chapter is organized as follows: Section 4.2 briefly introduces the two classifiers SC and MLP used in the probability combination. Section 4.3 explains the mathematical basis of the fusion of posterior probabilities aimed at generating a lattice as input for a Viterbi decoder. In Section 4.4 we give a detailed explanation and justification of the design of the experiments performed in this research. The results are presented in Section 4.5 and discussed in Section 4.6. The major findings are then summarized in Section 4.7.

4.2 The MLP and SC Classifiers

For the experiments in this chapter we use the AURORA-2 speech database, which contains sequences of up to seven connected digits from the set {*oh, zero, one,* $\cdots$, *nine*} corrupted by eight different types of additive noise at seven different noise levels (i.e. clean and SNR = 20, 15, 10, 5, 0, -5 dB) [11].

Although AURORA-2 is now becoming an outdated speech corpus for the purpose of state-of-the-art ASR word decoding, the effect of different features as presented in this chapter becomes more clear if there is no language model that might be a confounding factor during the word decoding. Without LM, WER improvement and deterioration can fully attributed to subtleties in the procedure to determine the acoustic features. We therefore considered this database adequate for giving a proof of principle that the merging algorithms proposed in this chapter constitute a valid approach.

As in most studies on AURORA-2, we model each digit as a sequence of 16 consecutive states (rather than, for example, a sequence of phone models). Silence is represented by a HMM model using three consecutive states. Therefore, all models in total comprise $(11 \times 16 + 3 =)$ 179 states.

In the front-end of our recognizer, we apply two different classifiers (MLP and SC). Each classifier produces a 179-dimensional posterior state probability vector, which is updated every 10 ms. Subsequently, as illustrated in Figure 4.1, the outputs of the classifiers are merged to yield a new stream of posterior probability estimates (also 179-dimensional vectors), which are processed by a Viterbi decoder back-end (implemented in MATLAB). The degree of success of a given fusion strategy (described in Section 4.4) is determined by evaluating the corresponding WER.

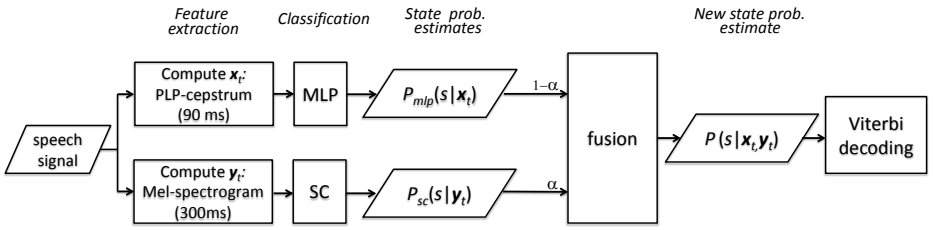


FIGURE 4.1: The state probability vectors of two different classifiers are merged to yield a new estimate, which is then used as input for a classical Viterbi decoder. The relative stream weights can either be chosen constant for an entire utterance (reflecting different, SNR-dependent classifier qualities) or dynamic (reflecting different local trustworthiness of classifier output frames, e.g. due to local SNR variation). The different fusion schemes are elaborated in Fig. 4.2.

The differences between the two classifiers are substantial. The MLP is a discriminative classifier, which has been widely used for acoustic modeling as an alternative for the Gaussian mixture model (GMM) [137]. Due to the discriminative nature of the training, the output vectors of an MLP classifier tend to attribute most of the probability mass to a single state. The MLP system used here was trained using the Quicknet software [138]. Its input vectors were created by stacking nine neighboring frame vectors (spanning a 90 ms time window). Each 39 dimensional frame vector consisted of 13 Perceptual Linear Prediction (PLP) cepstral coefficients ($c0 - c12$) and their corresponding first (Δ) and second order ($\Delta\Delta$) time derivatives. For building the MLP system, the multi-condition training set in AURORA-2 was split into a set of 7685 utterances for optimizing the MLP parameters and 755 utterances for cross-validation. The MLP had one hidden layer, the optimal size of which was determined based on the frame accuracy obtained on the cross-validation set.

By contrast, the SC system approximates 300 ms wide (30 frames) Mel-band energy spectrogram representations of speech segments as a sparse, non-negative, linear combination of exemplar spectrograms with a duration of 300 ms. The exemplars are taken from two dictionaries (one consisting of 8000 speech exemplars randomly extracted from the set of 7685 utterances for training the MLP system, the other of 4000 randomly selected exemplars of the noise regenerated from the multi-condition train set, plus 23 artificially created one-band exemplars [69]). The same set of 755 utterances from the multi-condition train set that were used for cross-validation during MLP training were set aside as a development set in case parameters in the Viterbi back end needed to be tuned to the different statistical properties of the posterior probability estimates of the SC classifier (or those of the merged MLP and SC streams).

The SC system that we applied is described in detail in [69]. For this chapter it is important to know that all frames in a speech exemplar are labeled as pertaining to one of the 179 states from the 16-state digit models or the 3-state silence model. Using the weighting coefficients of the speech exemplars found in the linear decomposition (i.e., the speech activation scores), each frame in a segment of speech input can thus be associated with a vector of posterior state probabilities. In practice it appears that this approach leads to the probability mass being distributed over more than one element of the output vectors of the SC classifier [139].

In the following we will denote the posterior probability estimated by the MLP system for state s_k $(k = 1..179)$ at time frame t by $p_{mlp}(s_k|\boldsymbol{x}_t)$, where $\boldsymbol{x}_t$ denotes the feature vector at time t. Similarly, the posterior probability estimated by the SC system for feature vector $\boldsymbol{y}_t$ is denoted by $p_{sc}(s_k|\boldsymbol{y}_t)$. The PLP-based frames $\boldsymbol{x}_t$ in the MLP system and the energy spectrum based frames $\boldsymbol{y}_t$ in the SC system are strictly synchronized in time. We will use different symbols $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$ throughout to emphasize that the posterior probability estimates are based on different acoustic features. When the classifier is irrelevant or clear from the context we will drop the subscript *mlp* or *sc*.

Since $p_{mlp}(s_k|\boldsymbol{x}_t)$ refers to the probability associated to a single state s_k, we will denote the 179-dimensional probability vector for all states by $p_{mlp}(s|\boldsymbol{x}_t)$ (and analogously for the SC by $p_{sc}(s|\boldsymbol{y}_t)$).

4.3 Weighted Stream Combination

4.3.1 Viterbi Decoding

In order to explain the relevant issues when combining multiple streams, we start from the well-known equations describing single stream Viterbi decoding:

$$\begin{aligned}\hat{W} &= \underset{W}{\operatorname{argmax}}\ P(W|\boldsymbol{O}) = \underset{W}{\operatorname{argmax}}\ P(\boldsymbol{O}|W)P(W) \\ &\approx \underset{W}{\operatorname{argmax}}\ \max_{s_t \subset W} \left(\prod_{t=1}^{T} p(\boldsymbol{o}_t|s_t)p(s_t|s_{t-1}) \right) \cdot P(W)^L \cdot e^{N_w \cdot \mathcal{W}_w + N_s \cdot \mathcal{W}_s} \end{aligned} \tag{4.1}$$

In the equation above, $\hat{W}$ represents the most likely word sequence across all possible word sequences W, given the sequence of observed acoustic feature vectors $\boldsymbol{O}$. s_t $(1 \leq t \leq T)$ indicates the state at time t along the hypothesized path W. The factor $P(\boldsymbol{O}|W)$ denotes the conditional probability of $\boldsymbol{O}$ given the hypothesis W, while $P(W)$ denotes the language model (LM) score of W.

In equation 4.1, the acoustic model (AM) factor $P(\boldsymbol{O}|W)$ is expanded in terms of $\boldsymbol{o}_t$ and $s_{k,t}$, which denote the observed acoustic feature vector and the HMM state $s_k(k = 1..179)$ at frame t $(1 \leq t \leq T)$, respectively. The expression $s_t \subset W$ is a shorthand for the set of sequences of states that form a word sequence hypothesis [in the AURORA-2 context a sequence of digits and silence (or background noise) tokens].

Because the AM and LM scores usually have substantially different ranges and different statistical behavior, a LM scale factor (L) is added to the equation as a power of $P(W)$ to balance the relative contribution of the LM scores to the AM scores. In addition, we have added a factor to Eq. 4.1 that includes two different penalties: a word insertion penalty for every digit (denoted $\mathcal{W}_w$), and a silence insertion penalty for every silence (denoted $\mathcal{W}_s$) in the hypothesized word sequence W (N_w and N_s denote the number of digits and silence tokens in the word sequence hypothesis W). These entrance penalties are added to the total path score upon entering a word or silence. In the decoding, these penalties balance the preference for decoding hypotheses with words that comprise shorter or longer stretches of speech and, in addition, the preference for digits versus silence tokens.

In ASR decoding schemes, usually only a single word insertion penalty is applied to every speech token in the hypothesis, while silence models are not penalized at all (silence is often dealt with in a special way). Pilot decoding experiments have shown, however, that in our case the separate balancing between word tokens and silence tokens in the decoding result is relevant to achieve an optimal weighting of different input streams. This is due to the combined effect of two mechanisms: (1) the word models consist of 16 states, while silence models consist of 3 states and (2) the MLP and SC classifiers used exhibit a substantially different behavior with respect to classifying silence (for example, the SC classifier appears to label relatively many silence frames as belonging to the begin and end states of digits). In actual practice, the insertion penalties cannot be chosen completely independently. The role of both insertion penalties will be discussed in the result Section 4.5.

Although from a theoretical point of view it is defensible to use posterior probability estimates $p(s_t|\boldsymbol{o}_t)$ for defining the lattice for a Viterbi search [140], it is known that recognition performance generally degrades if MLP outputs are used directly instead of $p(\boldsymbol{o}_t|s_t)$ in Eq. (4.1). In practice, it appears necessary to convert the posterior probabilities to likelihoods, taking into account the prior class probabilities ([141]; p. 181). Therefore, rather than feeding the Viterbi decoder directly with posterior probabilities, we compute scaled likelihoods $p(s_t|\boldsymbol{o}_t)/p(s_t)$ and use the following equivalent of Eq. (4.1) as the starting point of our experiments:

$$\hat{W} = \underset{W}{\operatorname{argmax}} \max_{s_t \subset W} \prod_{t=1}^{T} \left[\left(\frac{p(s_t|\boldsymbol{o}_t)}{p(s_t)} \right) \cdot p(s_t|s_{t-1}) \right] \cdot P(W)^L \cdot e^{N_w \cdot \mathcal{W}_w + N_s \cdot \mathcal{W}_s} \tag{4.2}$$

$$= \underset{W}{\operatorname{argmax}} \max_{s_t \subset W} \prod_{t=1}^{T} \left[\mathcal{L}(s_t) \cdot p(s_t|s_{t-1}) \right] \cdot P(W)^L \cdot e^{N_w \cdot \mathcal{W}_w + N_s \cdot \mathcal{W}_s} \tag{4.3}$$

in which $\mathcal{L}(s_t) = p(s_t|\boldsymbol{o}_t)/p(s_t)$.

This framework for processing the output of a single classifier is readily extended to accommodate multiple classifier outputs. In our case, the likelihood estimates produced by a single classifier are replaced by a weighted SUM or weighted PRODUCT of the SC and MLP classifier outputs. Thus, for each time frame t the likelihood of each state is obtained by:

$$\mathcal{L}_{sum}(s_{k,t}) = \alpha \cdot \mathcal{L}_{sc}(s_{k,t}) + (1 - \alpha) \cdot \mathcal{L}_{mlp}(s_{k,t}) \tag{4.4}$$

when using the SUM rule, or by

$$\mathcal{L}_{prod}(s_{k,t}) = [\mathcal{L}_{sc}(s_{k,t})]^{\alpha} \cdot [\mathcal{L}_{mlp}(s_{k,t})]^{(1-\alpha)} \tag{4.5}$$

when using the PRODUCT rule. In these formulae, $0 \leq \alpha \leq 1$ is a parameter weighting the relative contribution of each stream into the merged stream.

4.4 Stream Weighting Designs of the Experiments

In this section we discuss in more detail the different methods for fusing the posterior probability estimates from an MLP and an SC classifier for obtaining new acoustic likelihood scores $\mathcal{L}(s_t)$. The overall design of the set of experiments that we conducted is shown in Figure 4.1.

4.4.1 Frame Independent Weighting Using Oracle Knowledge of SNR

This is the first weighting scheme, in which we assume a frame independent weighting that depends solely on the stream and on SNR. Although it might be argued that the impact of the different noise types on the WER in AURORA-2 is at least as important as the impact of SNR, we started from the assumption that the noises in AURORA-2 are sufficiently stationary to justify that a single, static (i.e., frame independent) estimate of α that is solely dependent on SNR suffices to obtain (close to) optimal performance.

To avoid confounds from errors in the automatic estimation of the SNR per individual utterance, we decided to use 'oracle' knowledge about the actual SNR levels of the test utterances and to explore the effect of α on the WERs (averaged over the four different noise types) obtained on test sets A and B. Moreover, to ensure that the back end was well matched with the statistical properties of the merged streams, the WIPs ($\mathcal{W}_w$ and $\mathcal{W}_s$ from eq. 4.2) were tuned so as to optimize recognition performance on the development set (see Section 4.2) for a given stream weight α. Thus, the results of this experiment primarily serve as an estimate of the ceiling performance that is achievable under the condition that the noises do not differ substantially between themselves, and that the noises are sufficiently stationary to make it unnecessary to adapt α to fluctuating local SNR in an utterance.

As in Eq. (4.4) and (4.5), we constrained the sum of the weights of two streams to be equal to one for both the SUM and PRODUCT rule.

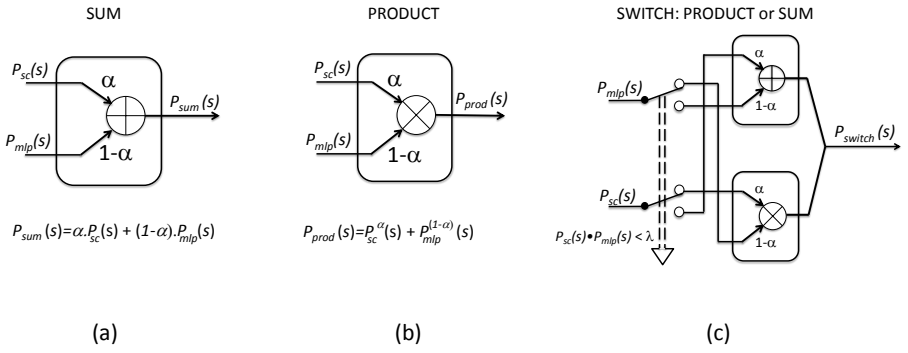


FIGURE 4.2: Overview of the three different weighting schemes to obtain a new state probability vector for each frame by fusing the corresponding classifier outputs. (a) SUM (b) PRODUCT (c) dynamic switching between SUM and PRODUCT based on level of agreement between classifiers (see text).

4.4.2 Dynamic Weighting Using Inverse Entropy

The assumption that using a fixed value of α, solely based on a global SNR estimate for a complete utterance, is (close to) optimal is likely to be too optimistic. After all, the local SNR might well change substantially, based on the unpredictable relation between the characteristics of a speech and a concurrent noise segment. Therefore, it might be profitable to make α *frame-dependent*, and determined by some characteristic of the state probability vectors that expresses the relative confidence of the respective classifier outputs on a frame-by-frame basis.

4.4.2.1 Inverse Entropy Based Weighting

The first method for making α variable on a frame-to-frame basis is the approach in which weights are based on the inverse entropy of the two probability vectors, in line with the approach presented in [50, 142]. The assumption underlying the inverse entropy weighting is that the fewer states take more of the probability mass (i.e., when the entropy approaches zero), the higher the 'trustworthiness' of this classification will be, and therefore the higher the weight for this probability vector should be in the merge. The inverse entropy weighting scheme favors, for each instant, the vector with state probability estimates from the stream with the lowest frame-entropy.

To introduce a quantity of which the value range is no longer dependent on the dimensionality of the classifier outputs, we first normalized the entropy $H(\boldsymbol{p})$ of

our 179-dimensional probability vectors $\boldsymbol{p}$ (with components $p(s_k)$) to fall within the interval $[0, 1]$ by dividing by $\log(179)$:

$$H(\boldsymbol{p}) = \frac{-\sum_{k=1}^{179} p(s_k) \cdot \log(p(s_k))}{\log(179)} \tag{4.6}$$

Then, setting α equal to the relative inverse entropies of the MLP and SC posterior vectors, makes the weighting in Eqs. (4.4), (4.5) (also see Figs. 4.1 and 4.2) frame-dependent:

$$\alpha = \frac{1/H_{sc}}{1/H_{mlp} + 1/H_{sc}} = \frac{H_{mlp}}{H_{mlp} + H_{sc}} \tag{4.7}$$

At each time instant, the parameter α weights the frame in the SC stream, while $1 - \alpha$ weights the frame in the MLP stream. Observe that the normalization in Eq. 4.6 does not impact the result in Eq. 4.7, thanks to the fact that both streams have equal dimension.

4.4.2.2 Trustworthiness Based Weighting

The inverse entropy weighting scheme has shown to be successful in experiments in which the posterior estimates came from two MLP classifiers that operated on different acoustic features [50, 142]. As argued above, the $1/H$-approach is based on the assumption that the inverse entropy is an appropriate estimation if the 'trustworthiness' of a posterior probability vector and that this scheme equally applies to all posterior probability vectors of all classifiers to be merged. In our experiments, however, this assumption is too simplistic. While the MLP vectors are characterized by a large posterior mass allocated to one single state, the SC system tends to attribute almost equal probabilities to multiple (often neighboring) states. Therefore, the entropy of the posterior vectors provided by the SC classifier are biased to have a relatively high value compared to the output vectors of the MLP classifier, even if the SC classifier is highly confident that a speech frame corresponds to some state or one of its close neighbor states that are acoustically very similar. Such 'confusions' between acoustically similar neighboring states will hardly affect the optimal path returned by a Viterbi decoder. That implies that the $1/H$ approach biases the stream with lowest entropy, also in the case in which the probability allocations in the corresponding frames are incorrect, that is,

irrespective of the intrinsic trustworthiness of the stream. Therefore, instead of considering $1/H$ as a measure for the trustworthiness of a stream, it would seem beneficial to base the trustworthiness estimate on the actual accuracy with which probability vectors predict the correct state label, and to find a classifier-dependent mapping from a posterior vector to a trustworthiness value.

For finding a mapping from stream-dependent entropy values to trustworthiness scores we followed the procedure proposed in [135]. In this procedure, the trustworthiness of a state-posterior vector is defined as the probability that the state with the highest posterior probability as assigned by the classifier is identical to the state assigned to that frame according to a golden standard.

To define the golden standard against which the winning states are to be compared, one has multiple options. For instance, it could be defined by a conventional forced alignment procedure, in which the golden state labels for each frame are the result of an alignment of the utterances with a corresponding HMM acoustic model sequence. Here, however, such a measure would be too strict (i.e., too conservative). The fact that all digits (including, for example, the digit *oh* that exhibits very little acoustic variation over time) are modeled as a sequence of 16 states leads to very similar acoustic characteristics of neighboring word-medial states. In addition, small temporal differences between the state segmentation in the reference and in the output of a classifier will in general have a negligible effect on the result of a Viterbi decoding. We therefore slightly relaxed the criterion for 'correctness' of the winning state in a probability vector, by taking into account the 'alignment neighbors' of a state in the definition of 'correct'. Since digit-internal states may be confused with their two neighboring states, a winning state N (with highest probability) is considered 'correct', even if it is classified as a direct neighbor $N-1$ or $N+1$. For winning digit-initial and digit-final states it holds that they are considered 'correct' if they correspond to the golden standard, to any of the silence states, or to the final state of any digit (in the digit-initial case) or the initial state of any digit (in the digit-final case). The distinction among three silence states is ignored as well.

In this scheme, the stream weight is computed as follows. In order to estimate the trustworthiness T of a state-posterior vector, first its normalized entropy H is computed. The trustworthiness of a posterior vector with normalized entropy in the half-open interval $[h1, h2)$ is defined by the proportion of all the vectors in

that entropy interval for which the 'winning' state is correct, using the relaxed definition of correctness described above.

Given this definition of trustworthiness T, we obtain a relation between normalized entropy and trustworthiness for each of the two streams (MLP and SC). This analysis was carried out on the development set (cf. Section 4.2). The relationship between entropy and trustworthiness is shown in Fig 4.3. In this plot, the normalized entropies are first quantized into 100 bins for each stream separately in such a way that each bin contains the same number of observations (each bin has therefore variable width). Next, the trustworthiness (i.e. the probability of correctness, according to the relaxed definition of correctness described above) is computed for all frames within each of these entropy bins. The results are shown in Figure 4.3 by red circles for the MLP and blue squares for the SC classifier.

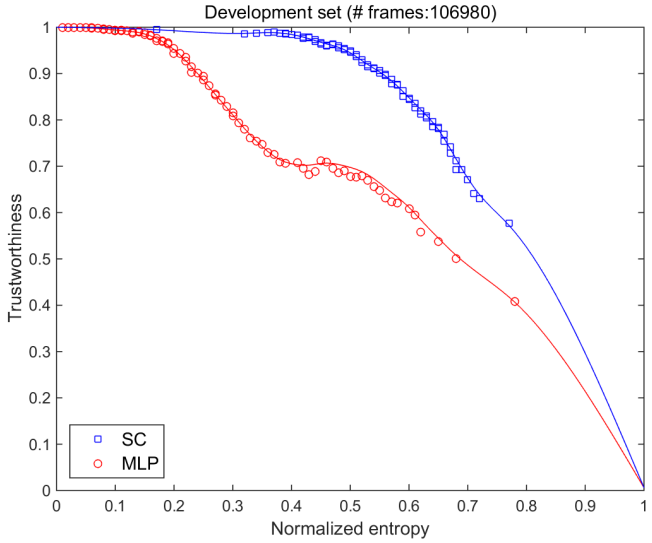


FIGURE 4.3: Trustworthiness T estimated on the 755 utterances in the development set as a function of normalized entropy H. Each marker symbol (red circle for MLP; blue square for SC) represents an equal number of frames (i.e., $106980/50 \approx 2140$). Smoothing spline approximations of the data points, which are used to derive the stream weights during recognition, are drawn as solid lines

The trustworthiness curves as shown in Fig. 4.3 deviate appreciably from a hyperbolic shape implied by $1/H$. This makes the actual trustworthiness as measured on the acoustic data, a potentially interesting alternative for the $1/H$ based weighting scheme discussed in the previous section. In addition, the figure brings to light two differences in the behavior of the MLP classifier and the SC classifier.

First, there are only a few samples in the SC stream for normalized entropy values < 0.4. This shows that the SC system hardly ever singles out one specific state.

This is in line with the earlier observations about SC and can be explained by the construction of the SC probabilities: the posterior estimates from SC are obtained from the (positive) weights with which exemplars from a dictionary of 8000 speech exemplars (with a state label assigned to each frame of the exemplar) must be added to approximate a segment of an unknown speech signal. Moreover, each posterior vector is based on the average of the state activations in up to 30 window positions. Using the 16-state digit models in the AURORA-2 task, this cannot but lead to very similar posteriors for neighboring states that are very similar in acoustical terms. This line of argument is closely related to the relaxation of the correctness constraint for the computation of trustworthiness.

Ignoring the extra hump in the MLP curve at $H \approx 0.5$ for the moment, the second difference between SC and MLP is that the SC curve has grossly speaking a similar shape as the MLP curve, but is shifted to the right. In other words, to achieve the same trustworthiness, the SC stream has a much higher entropy than the MLP stream, which is again in line with previous observations. In the middle range of the normalized entropy, the difference of the trustworthiness between SC and MLP can be up to around 0.3. Also the fact that the relation between entropy and trustworthiness depends on the stream, suggests that using a stream-independent measure such as $1/H$ is too coarse.

In order to address this difference, we conducted an experiment in which we replaced the $1/H$-based stream weights by weights based on the trustworthiness estimations T derived from the smoothing spline approximations that are drawn as solid lines in Figure 4.3, applying Eq. 4.8. This will be discussed in the next section.

$$\alpha = \frac{T_{sc}}{T_{mlp} + T_{sc}} \tag{4.8}$$

4.4.3 Dymamic switching between SUM and PRODUCT Rules

In [88] we found that the PRODUCT rule for fusing the two classifiers yields lower WERs than the SUM rule in the higher SNR conditions, but that the SUM rule outperforms the PRODUCT rule in the lower SNR conditions. This observation is in line with [132, 133] who already pointed out that it is advantageous to apply the SUM rule if the classifiers to be merged considerably disagree, which is, as we observed, a condition that becomes more likely when the SNR decreases.

This finding inspired us to investigate methods for *dynamically switching* between the two combination operations (SUM, PRODUCT) depending on some estimate of the level of agreement between the two classifiers. In order to quantify this agreement, we adopted the scalar product (dot product) between the two (SC and MLP) classifier probability vectors at each time frame t as a measure for the agreement (denoted A_t) between the two vectors:

$$A_t = \sum_k P_{sc}(s_k|\boldsymbol{x}_t) \cdot P_{mlp}(s_k|\boldsymbol{y}_t) \tag{4.9}$$

Note that if both vectors consist of probabilities of two classifiers with vector sum equal to 1, A_t denotes the probability that the two classifiers agree about the same estimate (a.k.a. probability vector direction) [143, 144]. In order to be able to apply different rules dependent on the level of agreement, we employed a threshold λ in such a way that if $A_t < \lambda$ the SUM rule is applied, and the PRODUCT rule otherwise. This 'dynamic switching' is schematically depicted in Figure 4.2c.

As in the recognition experiments described above, the weights of the SC and MLP stream were estimated on a frame-by-frame basis from either the inverse entropy $1/H$ (eq. 4.7) or from their estimated trustworthiness T (eq. 4.8). The decision threshold λ was optimized on the development set, in a 2-step procedure. In the first step a coarse grid search ($0 \leq \lambda \leq 1$; step size 0.1) was done. This was followed by the second step using a finer grid in in the region of interest ($0 \leq \lambda \leq 0.1$; step size 0.01). A value $\lambda = 0.05$ as the optimal threshold obtained on dev-set both for the inverse entropy and the trustworthiness weighting method. This value was then used in all subsequent experiments.

4.5 Results

4.5.1 Frame Independent Weighting Using Oracle Knowledge of SNR

As explained before, the main purpose of this experiment is to see to what extent recognition performance can be improved by a static (i.e. frame independent), weighted fusion of the state posterior estimates from the SC and MLP classifiers

while the stream weights are based on oracle knowledge of the SNR. To that end, we selected the (static) weights w_{sc} from the set $\alpha = \{0, 0.1, \cdots, 0.9, 1\}$; the corresponding weight w_{mlp} was set to $1 - w_{sc}$. The penalties ($\mathcal{W}_w$ and $\mathcal{W}_s$) were tuned on the development set (see Section 4.2) for each value of α so that optimal recognition performance (averaged over the SNR conditions 0 to 20 dB) was achieved. The resulting WERs with fusion based on the SUM rule are shown in the upper half of Table 4.1; the WER obtained with the PRODUCT rule are shown in the lower half. The performance of single-stream MLP and SC systems correspond to the top (MLP) and bottom (SC) rows in the two sub-tables. Since with weights of zero and one there is no fusion, these rows are identical in the upper and lower half of the table.

The data in Table 4.1 confirm previous findings [139], but provide more detailed information about the impact of SNR. In test set A, the MLP-only classifier outperforms the SC-only classifier up to SNR 5 dB, while SC-only is (much) better at 0 dB and -5 dB. Apparently, with known noise types (the noise dictionary in the SC system is derived from the noises in test set A) the SC classifier does a better job than the MLP in generalizing to low SNR levels that were not part of the training conditions. This trend is also supported by the trend in the bold figures (showing the WER minima per column): the lower the SNR, the higher the required SC weight to achieve minimal word error rate. In test set B the MLP-only classifier is always superior to the SC-only classifier, suggesting that the SC classifier has difficulty to generalize to noise types that are not represented in its noise dictionary. From a comparison of the results in the upper and lower part of Table 4.1 it can be concluded that the PRODUCT rule performs slightly better in the highest SNR condition, while the SUM rule is to be preferred in the lower SNR conditions.

Perhaps the most interesting observation from Table 4.1 is that the maximum performance is never achieved in the single stream scenario (i.e. in the top or in the bottom row). Apparently, it is always beneficial to merge the two streams in some way, even in conditions in which one stream performs better than the other. For clean speech the WER for the MLP-only system equals 0.9% WER, whereas the WER for the SC-only system is 2.8%; still, fusion with adequate weights for both classifiers decreases WER to 0.7% (SUM rule) and 0.6% (PRODUCT rule), respectively. Similarly, in the -5 dB condition in test set A, WER=33.0% for the SC-only classifier, compared to WER=53.2% for the MLP-only classifier. Yet,

fusion - admittedly with a high weight of 0.9 of the SC classifier- yields a WER of 31.3% (SUM rule) and 31.7% (PRODUCT rule), respectively.

4.5.2 Dynamic Weighting

In this section we explore to what extent the dynamic stream weighting procedures proposed in Section 4.4.2 can compete with the static weighting approach described in the previous section. The WERs are summarized in Table 4.2. For ease of comparison and interpretation the top rows of the table repeat information that was already given in Table 4.1. The rows labeled *sc (base)* and *mlp (base)* repeat the results obtained with the single-stream SC and MLP classifiers only. The rows labeled *oracle(+)* and *oracle(×)* contain the best results at each SNR (bold figures) from the corresponding columns in Table 4.1. The columns labeled *0-20* contain the WER averaged over the SNR conditions between 0 and 20 dB. These columns are provided to enable quick comparisons with the literature which often only reports these average scores. The best WER in each of the columns is printed in bold.

The rows labeled *invH* show the results using inverse-entropy dynamic weights that were computed using Eq. 4.7; analogously, the rows labeled *trust* refer to results obtained with trustworthiness-based dynamic weights that were computed using Eq. 4.8. The symbols +, ×, and +/× indicate whether the fusion was done using the SUM rule, the PRODUCT rule, or the dynamic switch between the two, respectively.

Both with inverse entropy and trustworthiness, the weights were determined for each time frame, independently of surrounding time frames. The exact same weights were used with the PRODUCT and the SUM rules for fusing the posterior estimates produced by the two classifiers. In the experiments in which we switched between the PRODUCT an SUM rules the switch criterion was $\lambda = 0.05$ (see Fig. 4.2). In practice this means that the SUM-rule is only reverted to in case of a relatively large disagreement between the two classifiers.

The most striking message conveyed by Table 4.2 is that fusion is always beneficial, even if the accuracy of the individual classifiers may differ substantially. All fusion-based results in the *clean* condition are better than the result of the MLP-only classifier, and all fusion-based results in the -5 dB SNR condition are better than

TABLE 4.1: Oracle recognition results on AURORA-2 in terms of word error rate (WER) as a function of the frame-independent stream weights w_{sc} and w_{mlp}. The columns represent the different noise conditions present in AURORA-2; the rows relate to the different settings of w_{sc} and w_{mlp}. The upper panel deals with the situation in which the weighting scheme is SUM, while the lower panel relates to the PRODUCT weighting scheme. SC and MLP baselines (single streams) can be found when w_{sc} and w_{mlp} equals to 1, respectively. Bold figures refer to the minima within each column.

Sum rule			test set A						test set B					
w^{sc}	w^{mlp}	clean	snr20	snr15	snr10	snr5	snr0	snr-5	snr20	snr15	snr10	snr5	snr0	snr-5
1.0	0.0	2.8	3.6	4.4	6.0	9.6	17.1	33.0	3.7	4.6	7.5	14.1	29.3	60.6
0.9	0.1	1.6	2.0	2.4	3.7	6.8	14.2	**31.3**	2.1	2.6	4.4	9.9	23.6	56.0
0.8	0.2	1.1	1.4	1.8	2.9	5.5	12.7	31.4	1.6	2.1	3.4	8.2	21.3	53.0
0.7	0.3	0.9	1.3	1.5	2.5	5.0	**12.3**	31.9	1.3	1.7	3.1	7.5	20.3	51.2
0.6	0.4	0.8	1.1	1.4	2.5	**4.9**	12.4	33.2	1.2	1.6	2.9	7.1	20.0	**50.5**
0.5	0.5	0.8	1.0	1.3	**2.4**	5.0	13.0	35.1	1.2	1.6	3.0	**7.0**	**19.8**	50.6
0.4	0.6	0.8	**1.0**	**1.3**	2.4	5.1	13.7	37.1	**1.2**	**1.6**	**2.9**	7.1	19.9	51.4
0.3	0.7	**0.7**	1.0	1.3	2.4	5.4	14.9	40.5	1.3	1.6	3.0	7.3	20.6	52.6
0.2	0.8	0.8	1.0	1.3	2.5	5.6	16.4	44.2	1.3	1.8	3.1	7.5	21.3	54.1
0.1	0.9	0.9	1.0	1.4	2.7	6.1	18.4	48.7	1.4	1.9	3.2	7.9	22.5	56.3
0.0	1.0	0.9	1.0	1.5	3.0	6.9	20.9	53.2	1.4	2.0	3.6	8.9	24.3	58.7
Product rule			test set A						test set B					
w^{sc}	w^{mlp}	clean	snr20	snr15	snr10	snr5	snr0	snr-5	snr20	snr15	snr10	snr5	snr0	snr-5
1.0	0.0	2.8	3.6	4.4	6.0	9.6	17.1	33.0	3.7	4.6	7.5	14.1	29.3	60.6
0.9	0.1	1.4	1.9	2.3	3.7	6.7	14.2	**31.7**	2.0	2.7	4.7	10.3	24.5	56.3
0.8	0.2	1.0	1.3	1.7	2.8	5.5	12.9	32.2	1.5	1.9	3.6	8.6	22.3	53.2
0.7	0.3	0.7	1.1	1.4	2.5	5.1	**12.7**	33.8	1.3	1.6	3.2	7.8	21.1	51.8
0.6	0.4	**0.6**	0.9	**1.2**	2.3	**5.1**	13.2	35.7	1.1	1.4	**3.0**	7.6	**20.5**	**51.5**
0.5	0.5	0.7	0.9	1.2	**2.2**	5.1	13.5	38.2	**1.1**	**1.4**	**3.0**	**7.5**	20.7	52.1
0.4	0.6	0.7	0.9	1.3	2.3	5.3	14.7	40.9	1.1	1.5	3.1	7.6	21.0	53.0
0.3	0.7	0.7	**0.9**	1.3	2.4	5.5	15.7	43.9	1.2	1.6	3.1	7.8	21.5	54.1
0.2	0.8	0.8	0.9	1.4	2.5	6.0	17.2	47.0	1.3	1.8	3.4	8.0	22.2	55.2
0.1	0.9	0.8	1.0	1.4	2.8	6.5	19.2	49.7	1.4	2.0	3.5	8.4	23.2	56.9
0.0	1.0	0.9	1.0	1.5	3.0	6.9	20.9	53.2	1.4	2.0	3.6	8.9	24.3	58.7

TABLE 4.2: WER (in %) on the AURORA-2 task using different weighting schemes and combination rules. For comparison, the rows 'mlp base' and 'sc base' present the performance of the two single-stream baseline systems (SC, MLP). 'oracle' indicates the optimal performance at each SNR with the tuned oracle static weights found in Table 4.1. The rows 'inv' and 'trust' represent a system using the inverse entropy based or data-driven weighting scheme respectively. The symbols '+', '×' and '+/×' refer to the SUM, PRODUCT and the dynamic switch between those two combination rules, respectively. The relative word error rate reduction in % between 'inv(+/×)' and 'trust(+/×)' is shown in the bottom row [%WERR(+/×)].

		test set A (dB)							test set B (dB)						
	cln	20	15	10	5	0	-5	0-20	20	15	10	5	0	-5	0-20
mlp(base)	0.9	1.0	1.5	3.0	6.9	20.9	53.2	6.7	1.4	2.0	3.6	8.9	24.3	58.7	8.0
sc(base)	2.8	3.6	4.4	6.0	9.6	17.1	33.0	8.1	3.7	4.6	7.5	14.1	29.3	60.6	11.8
oracle(+)	0.8	1.0	1.3	2.4	5.0	**12.3**	**30.7**	4.4	1.2	1.6	3.1	7.4	**19.8**	**49.7**	6.6
oracle(×)	0.7	**0.9**	**1.2**	2.3	5.1	12.8	31.3	4.5	1.1	1.5	3.1	7.9	21.5	50.8	7.0
invH(+)	0.8	1.0	1.3	2.4	5.0	13.0	33.7	4.5	1.2	1.6	3.1	7.2	19.9	50.3	6.6
invH(×)	0.7	1.0	1.3	2.3	5.3	13.6	36.3	4.7	1.2	1.6	3.2	7.8	21.0	51.8	7.0
invH(+/×)	0.7	1.0	**1.2**	2.3	5.0	12.8	33.8	4.5	1.2	1.5	2.9	7.3	20.0	50.5	6.6
trust(+)	0.7	1.0	1.3	2.3	5.0	12.5	32.6	4.4	1.2	1.6	2.9	**7.1**	20.0	50.5	6.6
trust(×)	**0.6**	**0.9**	**1.2**	2.2	5.1	13.1	34.6	4.5	1.1	1.4	3.0	7.5	20.7	51.3	6.7
trust(+/×)	**0.6**	**0.9**	**1.2**	**2.1**	**4.9**	12.5	32.7	**4.3**	**1.0**	**1.3**	**2.8**	**7.1**	19.9	50.5	**6.4**
%WERR(+/×)	14.3	10.0	0.0	8.7	2.0	2.3	3.3	3.1	16.7	13.3	3.4	2.7	0.5	0.0	2.4

the result of the SC-only classifier in test set A. Despite substantial differences, both classifiers seem to be good enough to always contribute useful complementary information.

From Table 4.2 it can also be seen that all systems that apply some form of dynamic weighting obtain accuracy scores that are very close to the scores obtained in the oracle experiment. This holds both for the inverse entropy weighting and the trustworthiness weighting procedures.

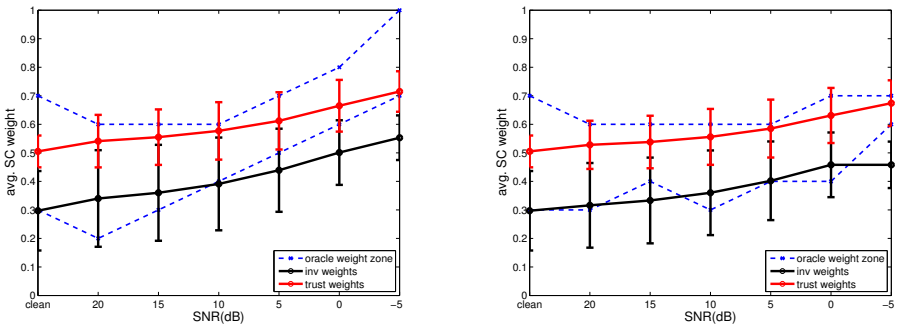
To facilitate comparison of the dynamic weight values with the static weights that yielded the lowest WERs in Table 4.1, we have depicted the mean and standard deviations of the weight values of the SC stream (i.e. α of Eq. 4.7 and 4.8 respectively) per SNR for test set A and B in Fig. 4.4a and 4.4b, respectively. The weight values using the inverse-entropy approach are depicted in black; the corresponding trustworthiness-based weights in red. The two dashed curves demarcate the weight region that corresponds to the 95% confidence interval around the maximum performance levels of Table 4.1.

From Figure 4.4a and 4.4b it is clear that, regardless whether the weights are derived using the inverse entropy or the trustworthiness approach, the dynamically computed weights follow a similar trend as the SNR-dependent oracle weights. Interestingly, the bulk of all weights derived with the trustworthiness approach fall more precisely within the region that is demarcated by the dashed lines than those of the inverse entropy weights. Thus, both entropy based procedures for dynamically estimating the weights appear a viable alternative for an error-prone estimate of the utterance-based SNR.

4.6 Discussion

4.6.1 Dynamic vs. Static Weights: Advantages of using Local Information

From the 'cheating' experiment in Section 4.4.1 where the stream weights were tuned on the test data using oracle knowledge about the SNR, it became clear that virtually any weighted combination of the SC and MLP posterior probability streams yields a better recognition performance than their single stream counterparts alone (cf.



(A) test set A (B) test set B

FIGURE 4.4: *Means of the dynamic weights (error bars indicating ±1 standard deviation) per SNR using the inverse entropy (black) and the trustworthiness (red) approach. The two dashed curves demarcate the weight region yielding a performance which is within the 95% confidence interval around the maximum performance of Table 4.1.*

Table 4.1). However, the table also shows that −to obtain the the best achievable performance at every SNR− the stream weights must be made SNR-dependent. This implies that choosing a single, SNR-independent weight inevitably causes a trade off between high and low SNR conditions. To adapt the stream weights to different noise levels automatically, two different methods were investigated for adjusting the stream weights dynamically, i.e., on a frame-by-frame basis.

Both methods hinge on the very same idea: in a given state probability vector produced by a certain classifier, the likelihood of the state with the highest probability being correct is closely tied to the entropy of that vector. Silently assuming that (inverse) entropy of a classifier output frame reflects its trustworthiness irrespective of the underlying classification principle, our first method assigns the weights directly proportional to the inverse-entropy of the state probability vectors produced by the MLP and SC classifiers (cf. eq. (4.7)). Subsequently, realizing that the MLP and SC classifiers yield output vectors with substantially different entropy ranges thereby potentially violating the previously mentioned assumption, we introduced a second method in which a development set was used to derive an empirical mapping from the entropy of the state probability vectors to frame-based "trustworthiness". The latter was defined as the proportion of the classifier output frames with a given entropy in which the most likely state label appeared correct according to some pre-defined golden standard (forced alignments).

Table 4.2 illustrates that both the inverse entropy and the trustworthiness weighting yield recognition performance levels which are similar, or in some cases even slightly

better, than the ceiling performance that was obtained by tuning static weights per SNR condition on test sets A and B. In addition, the trustworthiness weights further improve recognition performance over the inverse entropy weights at almost all SNRs.

Generally speaking, the latter finding confirms that it is sub-optimal/incorrect to consider (inverse) entropy as an absolute confidence measure that remains valid across different classifiers. Rather the relative weights of the classifier outputs should be determined in such a way that differences in entropy ranges are properly accounted for. Clearly in Fig. 4.4, the bulk of the "trustworthiness" weights (means including the ± 1 sd error bars) fall more precisely within the region that is demarcated by the dashed lines than the inverse entropy weights. This finding also forms a plausible explanation for the better performance of the 'trust' over the 'invH' performance in Table 4.2.

4.6.2 Use of Entropy and the Data-drive Approach

In this study, noise robustness is the central topic and it is commonly accepted that randomness of the classification results is proportional to the loudness of the noise. Therefore, entropy, as the universal physical term for randomness, is used to estimate the SNR and correspondingly suggests the combination weights. Figure 4.4 reveals that the mean weights from both the inverse entropy and the trustworthiness approach, increase monotonously with decreasing SNR. The fact that the slopes of the black and red lines in Figure 4.4 follow the trend of the region demarcated by the dashed lines more closely, is not really surprising. The similarity of the lines in Figures 4.4a and 4.4b indicates that the dynamic weights mainly reflect the SNR, for both test set A and B. Apparently, entropy in both methods can serve as an effective way to account for different noise levels. In the opposite, however, if the speech recognition task is not involving different SNRs or the classification output is not entropy-sensitive, this proposed method will not help much. Even though, this idea can still be applied with an alternative criterion instead of entropy.

Another reason why using entropy is beneficial is the big distinction of SC and MLP estimates. Figure 4.3 shows the big portion of SC frames have an entropy between 0.4 and 0.7, which has only a little overlap with the MLP one locating below 0.4.

This fact obviously leads to the advantage of using trustworthiness-based approach where an independent confidence measure is done for each stream. Clearly, in Figure 4.3 the bulk of the "trustworthiness" weights (means including the ±1 sd error bars) fall more precisely within the region that is demarcated by the dashed lines than the inverse entropy weights. This finding also forms a plausible explanation for the better performance of the trust over the 'invH' performance in Table 4.2. If the difference is less obvious, the benefit of such method would not be necessarily promising and possibly a unique measure such as inverse entropy approach would be good enough.

Additionally, another observation to be made from Table 4.2 is that the recognition performance for test set A at SNR=-5 dB is slightly below the level obtained with the oracle weights. Also the trustworthiness weights in this condition are a little low compared to the oracle weights between the dashed lines (cf. Figure 4.4). This can be explained by the intrinsic property of data-driven approaches: they normally cannot compensate for a lack of generalization power of the classifiers that becomes apparent when noise types are presented that were not available during training. We believe this can be attributed to the fact that the development data did not contain the noisiest conditions (SNR=0 and -5 dB). Presumably, it is this lack of data for the high entropy end of the scale in Figure 4.3, which makes the table lookup procedure less accurate in assigning the proper weights to frames with a high entropy.

4.6.3 Confidence Measure: State Correct Rate at the Frame Level

Different from the correctness at the phone level used in [50], the word-state-level correctness is used in this chapter. This much finer judgment (the total amount of classes is increased from 20 to 179) on the one hand provides a more accurate evaluation of the quality of classification, on the other hand, is more sensitive to "mistakes" which might not matter. For instance, each digit is modeled 16 states in this work. The boundary states (state 1 and 16) shows a very similar acoustic property as silence ones. In this case, it becomes harmful to make a too strict decision at the state level. This is the reason why we merged all boundary states together with the silence ones and ignoring differences between neighboring states while evaluating the correctness in Section 4.4.2.2.

In this work, we used the frame-wise correctness to estimate the trustworthiness, because of the easiness of calculation. However, the averaged dynamic weights, according to either the frame-wise inverse entropy based or the frame-wise data-driven approach, result in a WER which is very close to the one with a universally static scheme of the similar weight, shown in Figure 4. It suggests that the Viterbi decoder might be robust enough to smooth out the variations introduced by the frame-wise approach. Therefore, although it is shown in [103] that using a certain amount of temporal context may help obtain a more accurate estimate of the stream weights, whether it will bring more improvement is not expected.

Moreover, it is somewhat crude to compare the winning state index with the label in the forced alignment and judge whether one frame is correct or not. First of all, the state boundary is quite blurred. Even though we introduced several exceptions in Section 4.4.2.2 to make the comparison more fair, one may still come up with many ideas to improve the criterion. In this work, we mainly would like to emphasize the importance of an analysis for each single stream in order to measure the trustworthiness separately. More decent criteria are expected to be explored in the future and are likely to bring extra improvement.

4.6.4 Dynamic Switch between Combination Rules

Besides the promising results we obtained by combining MLP and SC streams together, we also studied how to avoid a contamination when one stream is much worse than the other. The hypothesis is that two streams are unlikely to make the same mistake given the same input. In other words, if the similarity of two estimates is high, it is likely that both are correct; otherwise at least one classifier breaks down. Given the fact that PRODUCT rule performs better than the SUM rule when both classifiers provide reasonable estimates and the opposite holds if one estimate is much worse, we introduced a novel algorithm which allows a dynamic change of the combination rule at the frame level. The scalar product of the probability vectors of each frame is used as a simple cost function to measure the similarity between two vectors.The very small threshold (0.05) for switching between SUM and PRODUCT suggests that only when the estimates from two streams are very different with each other (degree of similarity below 5%), a switch should be applied. In those scenarios, it is likely that only one of the two classifiers collapses while

the other one can still producing reasonable classifications. Therefore, a more conservative SUM rule (also can be interpreted as an average rule) is more in favor.

This switching mechanism is very efficient and effective, resulting in most of the best numbers in Table 4.2 without adding much computational complexity. Moreover, this approach is also easy to be extended to multiple (more than two) stream combinations.

4.7 Conclusion

In this work, we investigated how to effectively combine two ASR systems at the probability level. Instead of using a static weights over time, dynamic weighting schemes are introduced to be automatically adaptive to different SNR levels. A novel trustworthiness based approach, a data-driven one, is proposed aiming to be robust against the diverse properties among each individual stream. Experiments on AURORA-2 confirms the improvement of the new approach at a wide range of SNRs. Furthermore, we also compared the different combination effect between the SUM and the PRODUCT rule. In order to harness the strengthens of both rules, we proposed a dynamic switch between two, based to a very simple function – scalar product – to measure the similarity of two streams. This switch is shown to be robust not only at our new trustworthiness-based combination, but also at inverse entropy method. In the end, our final system, including the trustworthiness-based combination rule and the SUM and PRODUCT rule switch, not only performs better than each stand-alone streams, but also reaches the same performance as the oracle system where the combination weights are tuned for each SNR condition, without knowing the SNR.

In the future, several ideas are waiting to be investigated. For example, a more accurate way to estimate the trustworthiness of the estimated stream can be developed; a more smooth transition between the SUM and the PRODUCT rule can be allowed instead of a hard decision; a more advanced confidence estimator can be studied to further improve the combination. Additionally, investigations can be extended to more-than-two-stream combinations and large vocabulary tasks. All are potentially promising approaches to further improve this combination system.

Chapter 5

Off-line Lattice Combination with Dynamic Weights on Large Vocabulary Continuous Speech Recognition Tasks

5.1 Introduction

In Chapter 4 it was shown that when fusing information of several systems dynamic weighting, with weight determined by the trustworthiness of the streams, outperforms static weights. The experiments in that chapter were done on the AURORA-2 task, which allowed us to factor out the role of the language model. A trustworthiness based dynamic weighting scheme was proposed for fusing streams at the probability level, in a situation in which SNR conditions varied from -5 dB to 20 dB. Importantly, a stream that computed probabilities with Multilayered Perceptrons (MLP) outperformed a stream based on Sparse Coding (SC), while the SC-based stream outperformed the MLP-based stream at low SNR levels. The proposed dynamic weighting scheme managed to harness the strengths of both MLP and SC facing the large variety of noise conditions.

In this chapter, we investigate the benefit of dynamic weighting with a Large Vocabulary Continuous Speech Recognition (LVCSR) task with data recorded in operational applications of cars from five different brands. Due to the large range

of conditions encountered by the operational in-car ASR systems, such as various noise conditions caused by different driving speeds, road conditions, windows open or closed, presence of background music or a second speaker, the size of the vocabulary and the complexity of the language model, it is clear that the real-life applications must deal with a larger and less-controlled range of variations than what is typically found in carefully collected data sets such as AURORA-2.

It is reasonable to assume that different ASR systems that use different acoustic models will show differences in their robustness against specific adverse conditions. In addition, several parameters in the decoder, such as the word insertion penalty, Language Model factor and the depth of the search beam, can be tuned to optimize performance in certain conditions. Decoding parameters optimized for (relatively) clean speech may be detrimental to the performance in noisy conditions. Therefore, it is interesting to investigate whether fusing different operational ASR systems can increase overall performance and –specifically – whether here too dynamic weighting can be applied to advantage.

Instead of a confidence look-up table (see Figure 4.3), a dedicated MLP-based confidence model is trained to obtain the confidence scores per system in this chapter. The estimated confidences are used as dynamic weights in the system combination similar as Chapter 4. The proposed dynamic weighting approach is applied in a system combination with multiple acoustic models consisting of state-of-the-art deep neural networks which are trained independently, including Feed-forward Deep Neural Networks [145, 146], ResNet [147, 148], as well as uni- and bi-directional Long Short-Term Memory (LSTM) networks [149–151].

Lattices that combine the prediction of both AM and LM contain more information about the final recognition hypothesis. Therefore, we decided to investigate the impact of dynamic weighting when fusing word lattices. Two widely used lattice combination methods have been proposed in the literature: Confusion Network Combination (CNC) [52] and Minimum Bayes Risk (MBR) decoding [23, 152, 153]. In [23], the authors claim that MBR is superior to CNC from a theoretical point of view. They compared the two fusing methods and found that MBR consistently outperformed CNC in their HTK-based system; however, when fusing lattices generated by the IBM ASR system MBR did not outperform CNC. The authors called for additional experiments to shed more light on the issue, if not to settle it. In this chapter, we compared CNC and MBR with the proposed dynamic

weighting scheme. In addition to fusing pairs of ASR systems, we also investigate an extension to fusing multiple systems.

The rest of the chapter is organized as follows: Section 5.2 briefly introduces the LVCSR task and individual systems to be combined. Section 5.3 explains how confidence models are established for each component system and reviews the two lattice combination techniques used in this chapter. The results are presented in Section 5.4 and discussed in Section 5.5. Finally, the major findings are then summarized in Section 5.6.

5.2 Description of the Task and Baseline Models

5.2.1 Task Description

To verify the effectiveness of the confidence-based dynamic weighting scheme, we extended from the AURORA-2 digit task to a LVCSR task of Mandarin. We investigate the combination of five ASR systems that differ in the way in which the acoustic models are defined. In all systems the AMs are trained with the same data: 13k hours in total, consisting of two thirds of collection data and one third of field data from the in-car domain. The ASR systems also share the same language model that consists of an n-gram and a recurrent neural network component, targeted at the domain of in-car large vocabulary speech recognition. All systems and combinations are tested with field test data from five car OEMs. These data reflect the actual usage of the ASR systems, without any restrictions in terms of gender or age of the speaker, driving conditions, noise level or content of the utterances, etc., which would typically be balanced in collection data such as AURORA-2.

5.2.2 Acoustic Features

The acoustic features used to train deep learning acoustic models (AMs) comprise 45 Mel-frequency Cepstral Coefficients (MFCCs) and 7 fundamental frequency variations (FFV) features [154], extracted at a frame rate of 10 ms and a window size of 25 ms. The FFV features are added to capture variations of tones in Mandarin.

Additionally, a speaker-adaptive 100-dimension i-vector [155] is concatenated with the acoustic features as input for training the AMs.

5.2.3 Individual Model Description

Five large-scale ASR systems are trained as component systems for system combination. Two different data sets were used to optimize parameters in the decoder. One tune set comprised a wide range of noise conditions; the other set was dominated by recordings with a relatively high signal-to-noise ratio.

- First, a conventional feed-forward DNN (A:FF-DNN) is trained with the right and left contextual inputs of 7 frames. This FF-DNN model contains 6 hidden layers, all of which use the rectified linear (Relu) activation function. The output layer consists of a softmax activation function.
- The second model is a Residual Neural Network known as ResNet (B:ResNet). ResNets were first introduced for image recognition tasks with networks that contained a very large number of hidden layers to mitigate the problem with vanishing gradients. For that purpose the input of downstream layers consists of the sum of the previous layer and the output of a layer higher up in the architecture. The ResNet model used in this study has 10 hidden layers where the by-pass links are identity matrices established for the first 6 hidden layers over every 2 layers. The activation function for all hidden layers and output layer are also Relu and softmax activation function as in FF-DNN.
- The AM in the third system (C:FF-DNN#2) is identical to the one in A:FF-DNN, whereas the decoding parameters (such as language model factor, word insertion penalty and beam size) are optimized with the low-noise tune set. This configuration would be appropriate in high-end cars with silent air conditioners that would typically be driven with windows closed.
- The fourth model (D:LSTM) is uni-directional LSTM with 5 LSTM layers whose activation function is *tanh*. Three gates – an input, output and forget gate – regulate the information flow in and out of the memory cell. As a result, the network can take advantage of long time-contextual information. It also avoids the gradient vanishing and exploding problem that often plague vanilla Recurrent Neural Networks.

- The last model (E:bLSTM) is a bi-directional long short-term memory (bLSTM) recurrent neural network that can take advantage of long time contextual information in both past and future directions. The bLSTM used in this study has 6 hidden layers followed by a feed-forward bottleneck layer before the output layer. Activation functions for the LSTM layers are also *tanh*.

No time context is used in the input features in D:LSTM and E:bLSTM. Except for system #3 the decoding parameters were tuned to cope with a large range of noise conditions. It appeared that five systems obtained roughly similar performance within the test sets of the five OEMs. However, there were substantial differences between the five test sets, which most probably reflect differences in the overall use of the ASR systems in the cars of each brand. This makes it interesting to see whether a single fusion strategy exists that is (near) optimal under all usage conditions.

5.3 Off-line Lattice Combination with Dynamic Weights

5.3.1 Utterance-level Confidence Model

Instead of using the entropy of the posteriors as a confidence measure, a dedicated confidence model is trained for each of the 5 systems. The predictors are extracted from intermediate recognition results, including features such as LM scores, the number of surviving N-best results, the number of words, and average word duration. As all of these selected features are expected to reflect the recognition performance in the target testing domain to some extent, a simple logistic regression model is trained to estimate the confidence values. Binary labels are assigned at utterance level in training: 1 if no recognition errors at all, otherwise 0. The softmax function is used at the output layer to output confidence scores between 0 and 1, inclusive.

5.3.2 CNC and MBR combination

One popular system combination approach is ROVER [156], which uses the 1-best word sequences from multiple systems. The word sequences are aligned using a dynamic programming procedure. Different from the ROVER, the Confusion Network Combination (CNC) technique makes it possible to include a compact lattice representation, know as Confusion Network (CN) decodings, in the combination. A CN is a weighted directed sausage-like graph with a start node, an end node, and word labels over its edges. The CN has the peculiarity that each path from the start node to the end node goes through all the other nodes [157]. CNC allows alternative hypotheses to be taken into account by using CNs instead of 1-best word sequences in the DP alignment procedure [52]. It is shown that the use of word-level CNs and their corresponding probabilities improved the quality of combination performance significantly.

MBR-based system combination is also performed by re-scoring a set of likely hypotheses represented as lattices. The difference between CNC and MBR is that each node in a Confusion Network is associated with a time stamp and all combinations are restricted to hypotheses related to arc between consecutive time stamps. MBR-based combination does not have that time alignment constraint. It directly calculates a weighted Levenshtein distance between utterance hypotheses. The weights are derived from scores returned by the decoders. Compared to the light-weight CNC method, MBR is expected to allow more flexible lattice re-scoring. Since it is not confirmed that either CNC or MBR is superior in the literature [23], we compared both combination methods associating with various weighting schemes in this chapter.

The combination weights can be either static or dynamic. We used the confidence scores as dynamic weights per system in both CNC and MBR combinations. The confidence scores are at utterance level, so that the weights are adaptive per utterance.

5.4 Results

Five sets of field in-car data from different car OEMs are used to investigate the effects of static and dynamic weighting and possible differences between CNC

and MBR fusion. The test data reflect the most common applications of in-car ASR, such as messaging, voice control, control of the in-car music system and navigation. The baseline for characterizing the performance of the five ASR systems and the effects of system combination is the performance of the feed-forward DNN (`A:FF-DNN`), optimized for handling a wide range of noise conditions. The performance of the other ASR systems and of system combinations are expressed in the form of Character Error Rate Reductions (CERRs) relative to this baseline.

TABLE 5.1: CERR on the LVCSR task with CNC or MBR combinations. Static and confidence-based dynamic weights are compared.

			CNC			MBR
	#CHAR	B:ResNet	stcW(eq)	stcW(opt)	cnfW	cnfW
OEM1	633.1k	5.35%	4.67%	3.86%	6.77%	13.26%
OEM2	193.2k	0.67%	2.00%	2.60%	4.60%	10.00%
OEM3	133.1k	7.72%	8.33%	6.37%	12.01%	18.87%
OEM4	71.9k	2.63%	2.29%	2.97%	5.38%	11.44%
OEM5	4.2k	2.56%	2.56%	4.03%	5.91%	8.23%
average	207.1k	3.42%	3.65%	3.81%	6.53%	11.59%

Table 5.1 shows the results of an experiment in which we combined the reference system A:FF-DNN and ResNet DNN (`B:ResNet`) for the five test sets. The column `#CHAR` shows the total number of characters in the five test sets. It is evident that the sizes of the five sets differ substantially. The unweighted means of CERRs are given in the last row 'average' of each column which do not take the sizes of test sets into account. The numbers in the remaining columns show the percentage reduction (CERR) of system `B:ResHet` and different ways in which the `A:FF-DNN` and the `B:ResNet` systems are combined, relative to the baseline system A:FF-DNN.

To compare static and dynamic weighting schemes, three sets of weights are applied under the CNC umbrella: i) equal static weights 'stcW(eq)' (0.5 and 0.5), ii) optimized static weights 'stcW(opt)' and confidence-based dynamic weights 'cnfW'. First of all, it can be seen that `B:resNet` outperforms the reference system in all five test sets. The results in column 'stcW(eq)' in Table 5.1 show a compromise between two component systems `A:FF-DNN` and `B:ResNet`. If the numbers in this column are smaller than the corresponding numbers in the column B:ResNet, it means that the combination of the two systems performs worse than `B:Resnet` on

its own. This happens to be the case for OEM1 and OEM4. For OEM5 the CERR of the combination is equal to the gain obtained with `B:Resnet` alone.

The weights in 'stcW(opt)' are tuned from 0 to 1 with a step of 0.1 on a held-out tune set and the weight sum to one. The optimized static weights (0.6 and 0.4) are assigned to `A:FF-DNN` and `B:ResNet`, respectively. 'stcW(opt)' achieved three better-than-both CERRs in five test sets. However, in test sets OEM1 and OEM3 the CERR of the combination is still smaller than the gain of `B:ResNet` on its own, and the average CERR is only marginally better than 'stcW(eq)'. 'cnfW' with CNC shows substantial improvement over both static weighting schemes, where CERRs are nearly doubled consistently across all OEMs.

Applying constant, optimized static and dynamic weights with MBR showed similar results: dynamic weight is the only strategy that always yields a higher CERR than `B:ResNet` on its own. The rightmost column in Table 5.1 shows the results of combining `A-FF-DNN` and `B:ResNet` with dynamic weights. It can be seen that combination with MBR yields CERRs that are almost twice as large as the best results obtained with CNC.

Besides `A:FF-DNN` and `B:ResNet`, three more individual component systems described in Section 5.2.3 are evaluated and the corresponding CERRs over the baseline are shown in Table 5.2. The first three columns show that `C:FF-DNN2` and `E:bLSTM` outperform the reference system in all five test sets. However, `D:LSTM` performs worse than the reference in OEM2, and substantially worse in OEM5. It is striking that `C:FF-DNN2` the system with the decoding parameters optimized for relatively clean speech, shows the largest CERR relative to the reference of all four additional systems. This might suggest that realistic field data are less affected by adverse conditions than what is seen in collection data.

System combination results are shown in the right-hand part of Table 5.2, starting from column 'AB', from two-way up to five-way combinations. All combinations shown in the Table are based on MBR, which is shown to be superior to CNC in Table 5.1, associated with dynamic weights. The two-way combination 'AB' is identical to the column 'cnfW' with MBR in Table 5.1. Then one of the new systems `C:FF-DNN2`, `D:LSTM`, `E:bLSTM` is entered into the combination to forge three three-way combination 'ABC', 'ABD' and 'ABE'. Both involving D:LSTM and E:bLSTM provide a steady improvement over the two-way combination 'AB'. 'ABC' does not yield a better CER on average than 'AB', possibly due to the fact

TABLE 5.2: CERR on the LVCSR task with MBR-based lattice off-line combination.

	component systems			combined systems				
	C:FF-DNN2	D:LSTM	E:bLSTM	AB	ABC	ABD	ABE	ABCDE
OEM1	10.96%	9.61%	5.68%	13.26%	13.94%	16.98%	15.36%	17.93%
OEM2	4.67%	-5.00%	6.33%	10.00%	9.27%	10.00%	12.80%	12.27%
OEM3	15.44%	8.70%	3.43%	18.87%	19.24%	21.08%	20.22%	22.79%
OEM4	6.86%	3.20%	12.70%	11.44%	11.44%	15.68%	15.56%	17.05%
OEM5	3.63%	-29.02%	1.87%	8.23%	6.98%	6.98%	10.39%	9.54%
average	7.54%	-5.37%	5.44%	11.59%	11.28%	12.99%	14.07%	14.85%

that C:FF-DNN2 is too close to `A:FF-DNN`, so that not much new information is provided. Finally, all 5 systems are combined in 'ABCDE', yielding a 14.9% CERR over the reference system.

5.5 Discussion

5.5.1 Static vs. Dynamic Weights on LVCSR

We do not observe a big gap among the performances of individual systems in Table 5.1. The performance of five single systems in Table 5.2 shows that the worst-performing system `D:LSTM` differs relative 12% from the best one `C:FF-DNN2`. This situation is different from what is observed between MLP and SC in Table 4.2 in Chapter 4, where an absolute 20% difference exists at certain SNRs. One may argue that the dynamic weights would be less crucial here. However, neither of the two CNC combinations with static weights 'stcW(eq)' and 'stcW(opt)' in Table 5.1 show any benefits from the individual system on average. Only when the confidence-based dynamic weights are used, the avg. CER is improved significantly. This observation confirms that the proposed dynamic weights can adapt to test conditions based on the interaction among individual confidence estimates. It avoids compromised results which are obtained with static weights.

5.5.2 CNN vs. MBR

The results of our experiments show a consistent advantage of MBR-based system combination over combination on the basis of CNC. We refrain from an in-depth

analysis of the differences between the character sequences delivered by the two combination approaches. Therefore, we do not know whether our results support the theoretical claims about the superiority of MBR over CNC made in [23]. It may be that our decoder happens to be more similar to the HTK decoder than to the IBM decoder. Perhaps, the fact that the duration of spoken characters in Mandarin is short relative to the average length of words in English also plays a role.

5.5.3 Multi-stream Combination

Besides the significant gains, another advantage of using the confidence-based weights is that it becomes straightforward to integrate new systems into the combination as shown in Table 5.2. With static weights, the more systems involved, the more computationally complex the weight tuning would be. Usually the range of weights is between 0 and 1 inclusive with a step of 0.1. Then the grid search of the optimal weight set needs 11^N test sample points, where N is the number of systems to be combined. That is very expensive. Let alone, adding new streams means new weight tuning and the combination is only suboptimal since it is less-adaptive to any variations of testing conditions as described in the introduction. In Table 5.2, up to 5 systems are combined with MBR. No tuning of the weights is required since how the dynamic weights are obtained is estimated by the pre-trained confidence model.

It strikes the eye that the optimal three-way system combination does not include `C:FF-DNN2`, the system that on its own yields the largest CERR relative to `A:FF-DNN`. Equally interestingly, the three-way combination ABD consistently yields a higher CERR than ABC, despite the fact that `D:BLSTM` on its own always performs worse than `C:FF-DNN2`. Most probably, this is the result of the dynamic weighting that will promote the contribution of a system with a high confidence score, and demote systems with a low confidence score.

5.6 Conclusion

Experiments are done in this chapter to verify the effectiveness of the confidence-based dynamic weighting introduced in Chapter 4. On the one hand, the experiments use real-world large vocabulary data, instead of tightly controlled small vocabulary data. On the other hand, the number of component streams to be combined is extended up to five state-of-the-art deep neural networks. To measure the confidence of each individual system, a dedicated MLP-based confidence model is trained. Similar as in Chapter 4, the confidence scores are estimated at utterance level and utilized as dynamic weights to its own lattice in the off-line lattice combination based on either CNC or MBR. Combination based on MBR systematically outperforms CNC-based combination. The five-way MBR combination with dynamic weights provides the best performance of a relative 14.9% CERR over the FF-DNN baseline.

Chapter 6

Lexicon Study towards Accent Robustness in Mandarin ASR

6.1 Introduction

The term *accent* has various meanings, but in speaking, an accent is an identifiable style of pronunciation, often varying according to region or socioeconomic status. Recognizable accents commonly exist for most languages under the influence of local dialects. Although the only official language in both mainland China and Taiwan is Mandarin, almost a third of the population still do not speak what the government calls *putonghua* or the "common tongue" after over a century of promoting Mandarin as the official language of China. According to a recent Chinese government study, 400 million Chinese citizens cannot speak Mandarin. Of the 70% of the people who can speak Mandarin, many do not do it well enough. These persons speak one of over 1,500 dialects or heavily accented Mandarin [158, 159]. The standard Mandarin is based on the Beijing dialect.

Chinese dialects can be grouped, mainly on the basis of regional distributions (cf. Figure 6.1, adapted from [160]), and speakers of a dialect in one group may not be able to understand a speaker from a different group. Northern dialects in China tend to differ less from standard Mandarin than southern dialects. In addition, there are non-geographic factors that determine differences between dialects, such as the history and development of cities, as well as education level [161]. There is an important difference between dialects and accents: dialects are not limited to

FIGURE 6.1: Overview of the geographical distribution of major dialect groups in China. Adapted from [160]

specific pronunciations of certain words, but also include idiosyncratic expressions with words, phrases and even different grammatical constructs that may not occur in other dialects,. The definition of an "accent" in this paper is limited to variation in the pronunciation of words, in utterances that obey the standard grammar.

Accented speech poses a substantial challenge to Automatic Speech Recognition (ASR) systems; therefore, improving Mandarin accent robustness is a major step towards robust ASR [162]. Conventional approaches to improving robustness against accent differences were based on using larger training data sets that include different accent variations, or on using different types of speaker adaptations [163, 164], such as i-vector [165] or CMLLR [166, 167]. In this paper, we present an in-depth study of the effects of accent variation on ASR performance, with the goal of identifying the most powerful and cost-effective ways for improving the robustness of Mandarin ASR systems against accent variation in mainland China.

As mentioned above, accented data adhere to the standard grammar. Therefore, there is no need to adapt the language model in order to increase accent robustness.

In this paper we investigate two approaches to improving accent robustness. Firstly, we investigate how the AMs can be made more robust by modifications of the phone set. First, we merged all vowels with different tones to create toneless phones, aiming to build generic phone models that are robust against tonal variations in

the same vowel. Second, the toneless phones are added as a sixth tone to the original five tones per vowel. AMs are rebuilt with segmentation based on either the toneless phone set or the new extended phone set including six tones.

Secondly, we investigate how the lexicon can be extended so as to cope with accented pronunciation without changing the AMs. Previous studies [162, 168–170] have shown the effectiveness of extending the pronunciation model (PM) to cope with accented pronunciations. We investigate how we can determine phonetic confusions at different levels, such as context-independent tone confusions, context-dependent tone confusions and context-dependent syllable confusions that impact recognition accuracy most. In addition, a data-driven approach is developed to create the most promising confusion set automatically. The data-driven approach can easily be applied to a language for which we do not have detailed phonetic and linguistic knowledge.

6.2 Acoustic Modeling Approach

6.2.1 Baseline System

Mandarin Chinese is a tone language. Each Mandarin character corresponds to one and only one syllable, which obeys the consonant-vowel-consonant (CVC) structure [171]. In order to differentiate meaning, the same syllable can be pronounced with different tones. Mandarin's tones give it a very distinctive quality, but the tones can also be a source of miscommunication if not given due attention, especially when differences between accents play a role. Mandarin has four main tones and one neutral tone (or, as some say, five tones). Each tone has a distinctive pitch contour which can be graphed using the Chinese 5-level system [172].

Our baseline Mandarin ASR system distinguishes tones upon vowels, meaning that it treats the same vowel with different tones as independent phones, which results in five tonal phone units modeled for each vowel. For example, the vowel 'AA' is represented by five phones 'AA1', 'AA2', 'AA3', 'AA4' and 'AA5', where 'AA[1-4]' indicate the main four tones of the vowel 'AA'. 'AA5' indicates the neutral tone that occurs with substantially lower frequency than the other four tones, due to the fact that neutral tones usually only exist as the ending of certain phrases. All five tones have their own segmentations and are treated as independent phones.

Consonants have no tones attached. For example, there is only one consonant phone ‘B’ in ‘B AA1’ and ‘B AA2’. In total, the baseline phone set contains 159 mono-phones.

6.2.2 The Toneless Model

The main idea of the acoustic approach is to reduce the resolution of acoustic state modeling in order to make the models more tolerant to different variations of accented pronunciations. To achieve that, the phones in the original phone set that are most likely to be confused can be merged to be a new phone, which does no longer distinguish the confusable phones. For instance, it is well-known that the retroflex consonants such as ‘ch’, ‘sh’ and ‘zh’ may be replaced with dentals ‘c’, ‘s’ and ‘z’, respectively, in conversations usually in the South of China [173]. Therefore, the phonemes ‘ch’ and ‘c’, ‘sh’ and ‘s’, ‘zh’ and ‘z’ can be merged into three new phonemes which do not tell the difference between retroflex and dental consonants. Toneless models refer to the models which do not distinguish the five tones per vowel used in the baseline system. The motivation is to build models that are robust against tonal variations in accented Mandarin. For example, the toneless AM does not distinguish between ‘妈 (M AA1)’, ‘麻 (M AA2)’, ‘马 (M AA3)’, ‘骂 (M AA4)’ and ‘吗 (M AA5)’: all five vowels are mapped to a generic phone ‘AA’. This change is straightforward in practice: the time alignments in the baseline segmentation remains the same, while tone information is stripped from the original transcriptions. Correspondingly, the lexicon needs an adjustment to remove all tone-carrying variants from the lexicon. With this new toneless segmentation, the AM is re-trained.

6.2.3 The Sixth-Tone Model

The baseline system models all five tones per vowel in acoustic modeling, but that makes it less robust against non-standard tonal variations in accented speech. The toneless system, on the other hand, has better tolerances to tonal variations, but it sacrifices useful information about tone in its acoustic models. The sixth-tone model is a way to harness the strengths of the two approaches: it integrates the new toneless phones as a sixth tone for each vowel in the phone set. The idea is to achieve as good and precise modeling as the baseline system by the original five

tones per vowel, while at the same time providing a backdoor for tonal variations via the newly added sixth tone.

In practice, the sixth-tone model is trained with a mixture of tonal and toneless segmentations. One part of the audio data is annotated with the baseline segmentations, which only contain the original five tones. The rest of the audio data is annotated with toneless segmentations as described in Section 6.2.2, which only contain the toneless vowels. Tonal and toneless data are mixed at different ratios in order to find a good balance. Since the train set is split into tonal and toneless subsets, rather than duplicating audio data with new segmentations, the total amount of training data remains the same as what is used to train the baseline system. This procedure avoids the potential confusions in DNN training, if the same acoustic data are presented with multiple different labels. In this procedure the toneless model becomes a special case of the sixth-tone model, where only toneless segmentations are used in training, while the percentage of tonal segmentations is zero.

6.3 Lexicon Modification

The acoustic modeling approach, introduced in Section 6.2, has several disadvantages. Firstly, modeling generic toneless phones is an overshooting solution. Though accents introduce certain variations in pronunciations, not all possible confusions do occur, and those that do are limited to specific accents. Accent variations usually happen to certain words or expressions in specific contexts. While many accents might confuse AA_n and AA_m, it never happens that an accent confuses AA_n with all other tones. Therefore, it makes sense to investigate which tone confusions actually occur.

Secondly, accents are different from each other. However, the sixth-tone model, including the special toneless phones, cannot provide accent-specific solutions. The acoustic approach cannot customize the AM for a certain accent, if needed. Instead, it can only provide a compromised AM, which consists of all possible tonal confusions that may or may not occur in any accent.

Last but not the least, AM re-building is time-consuming and not adjustable. Usually, the effort needed to train new models is not an issue, because the AM is built offline and it only needs to be done once. However, given the fact that

there are many accents for bigger languages such as Mandarin, English, German, Spanish and French, the number of situations that might call for AMs that do account for specific accent might grow beyond reasonable training resources.

Instead of re-training the AM, we investigate a cheaper, but possibly more effective way to achieve accent robustness, while at the same time making the system more targeted at the phonetic variations that actually occur in specific accents. The lexicon acts as a bridge between the acoustic and the linguistic worlds. A modification of the lexicon can map the 'non-standard' AM prediction to a correct hypothesis.

The idea of a lexicon modification is that we can handle phonetic confusions by adding alternative pronunciations to the lexicon. For instance, if we observe that the first tone according in the standard pronunciation is always pronounced as the third tone in a specific accent, then the lexicon can be extended to contain all third-tone alternatives for all first-tone occurrences. Then a question becomes: how to define the phonetic confusion set for each accent? This can be done by linguistic experts for sure. In this study, we opt for a more general solution, assuming that we do not have advanced linguistic knowledge about changes from standard to accented pronunciations.

In the remainder of this section we explain all approaches to adapt the lexicon.

6.3.1 Alternative to Sixth Tone AM Rebuild: Full Extension of Tonal Variations

The idea of training a sixth-tone model is that we allow the generic phones to cover all tonal variations in accented speech. In theory, this is similar to expanding the pronunciations of all words in lexicon to all possible tonal variations. This is illustrated in the row of 'full extension' of Table 6.3.2, where V and m represent the original vowel and corresponding tone. Full extension of the lexicon allows alternative pronunciations with all other tones $V_{[1-5]}$.

This is a sanity check to link the acoustic approach in Section 6.2 and the lexicon approach here. Although the full extension does not support various balances between tonal and toneless representations, the lexicon approach does not require re-training of the AM.

TABLE 6.1: *Tonal confusion matrix*

REF\HYP	Tone1	Tone2	Tone3	Tone 4
Tone1	~~C11~~	C12	C13	C14
Tone2	C21	~~C22~~	C23	C24
Tone3	C31	C32	~~C33~~	C34
Tone4	C41	C42	C43	~~C44~~

Obviously, full extension of the lexicon over-generates, and might therefore negatively affect ASR performance.

6.3.2 Enumerate Tonal Confusions

Tonal variation is regarded as one of the most conspicuous phenomena of Mandarin accents. We limit the investigation to the four main tones, since the fifth (neutral) tone has much fewer occurrences than the other four. Table 6.1 shows the 4-by-4 matrix of possible confusions. Note that confusions are bi-directional, and not necessarily symmetric. We investigate all 12 possible confusions independently. For that purpose, we create 12 extended lexicons, and measure recognition accuracy on a development test set, without changing the AM. This allows us to identify the tone confusions that improve accent robustness most.

6.3.3 Typical Consonant Confusions

Consonant confusions are studied in a similar way as the tonal confusions. However, it is not feasible –and not necessary– to investigate all theoretically possible confusions. Instead, we created a shortlist of potentially relevant confusions, in part inspired by phonetic knowledge. The list is shown in Table 6.2. Note that here confusions are bidirectional and not necessarily symmetric. Note that, the diphthong confusions 'an↔ang, en↔eng, in↔ing' are categorized as ending consonant confusions.

TABLE 6.2: *A shortlist of consonant confusion pairs.*

initial consonants	zh↔z,ch↔c,sh↔s n↔l q↔x f↔h x↔s
ending consonants	an↔ang,en↔eng,in↔ing

6.3.4 Syllable-level Solutions

In this approach we investigate a set of more fine-grained confusions for each accent. For this purpose we study context-dependent confusions, combining tones and consonants, which were treated independently before.

We already increased the resolution once from Section 6.3.1 to Section 6.3.2, by limiting the expansion of the generic toneless rule to confusions between specific tone pairs. In this approach, we develop confusion rules at the syllable level. Instead of applying either tonal or consonant confusions in the lexicon globally, we now determine how ASR performance benefits from adding syllable-sized pronunciation variants to the lexicon. Only the syllable variants with a positive impact on ASR performance will survive and be applied in the final lexicon modification.

For this purpose we attach both of the original and the alternative phone sequences to the words in the lexicon. For example, the representation of the word '你好//ni3hao3' is modified as '你好//ni3hao3_ni2hao3', where 'ni2hao3' is the modified phone sequence resulting from the tonal confusion rule $C32$ in Table 6.1. We modified the decoder so that it outputs the preferred pronunciation variant along with the lexicon entry while processing the development set. This allowed us to identify the pronunciation variants that have a positive effect on CER. This procedure was carried out for all accents separately.

6.3.5 Accent-independent Solution

Different from retraining the toneless model or the sixth tone model, lexicon modification can be targeted at each accent individually. Although lexicon modification can be accent-specific, the lexicon enhanced for accent A may not be optimal for

accent B. Therefore, it would require an accent classifier, which is not trivial [164], to reap the advantage via lexicon enhancement. As a work-around, we investigated the intersection of accent-specific confusions and created an accent-independent confusion list that can be applied with all accents studies in this chapter. This is no different from treating all accented data as one super-set and we attempt to improve the overall performance via providing a universal lexicon solution. The overall performance does not only consider performances on heavy-accented, but also the standard and light-accented ones too. We tried this accent-independent solution at both tonal and syllable level.

6.3.6 Data-driven Approach

In the approaches described above some linguistic and phonetic knowledge was used in extending the lexicon. As a final approach, we use a fully data-driven and knowledge-free way to find the most promising extensions of the lexicon. For that purpose we compare the result of a forced alignment with standard Mandarin labels with the result of a free-phone decoding. This yields a matrix of actual confusions. We also create a confusion matrix based on recognition errors made by the baseline system when processing standard Mandarin. Both matrices were normalized, and we subtract the confusions in standard Mandarin from the confusions found in the multi-accent data. The outcome is supposed to be phonetic variation caused by the accent. For light accents, we would expect most of the large non-zero values to be on the main diagonal of the calibrated confusion matrix.The large values at off-diagonal entries will indicate the possible confusions we are looking for.

6.3.7 Trade-off Between Light/Heavy Accents and WER/RTF

The lexicon modification is done by adding alternative pronunciations, which increase the search space. This is helpful for recognition of heavily accented data that clearly differ from the standard. However, it also degrades recognition performance of light/standard speech and inevitably increases the real time factor (RTF). To find a good balance between recognition performance for light and heavy accented data on the one hand and RTF on the other, we limited the amount of added pronunciations into lexicon by adding only the N most frequent confusions found in the data-driven approach, where $\{N \in \{10, 100, 1k, 10k, 100k, all\}$.

6.4 Experimental Setup and Results

6.4.1 Accent Test Sets

The accented speech database used in this study contains 135 k read utterances (84.7 hours) from 468 speakers. The language is Mandarin as spoken in various regions across China. Recordings were made in 15 locations that were selected to obtain broad coverage of Eastern mainland China (see Figure 6.1). All speakers were born and raised in the respective dialect region; they speak the local dialect/language as their first language and learned standard Mandarin later. Number and gender of the speakers are balanced across accents (30 - 32 speakers per accent, half male and half female). Per accent the data was randomly split by speakers into train set(20 speakers)/test set(5 speakers)/development set(5-7 speakers). Details of the data set are summarized in Table 6.3.

The speech recordings originate from a scripted in-car human-machine interaction scenario. To create a realistic setting, the recordings were made in mid-size cars (various models per region) while driving on city roads and highways (approximately equal distribution of environments). This data collection setup ensures that differences between dialect groups are of acoustic-phonetic, not linguistic nature. All utterances are manually transcribed.

Each speaker was recorded in a single recording session that lasted about 15 minutes. Car audio equipment was turned off during all recordings and windows were closed. Less than 10 % of the recordings contain external noise such as wind, rain, etc. The data was recorded using far-talk and close-talk microphones; however, only the close-talk data is considered in this study, in order to eliminate the room impulse response of the car as a potential confounder. All data considered in this study were recorded with the same microphone model (Shidu S-43).

Each accent subset is expected to reflect the regional pronunciation habits, influenced by the local dialect. Therefore, it was expected that only the 'Beijing' set would contain close-to-standard Mandarin and all remaining locations would be characterized by some degree of accentedness. However, three trained native listeners who manually transcribed the recordings observed that only seven accents out of 14 (excluding Beijing accent) have clear differences in pronunciation from

Table 6.3: *Statistics of data collection.*

	accent	#SPK	#SNT	#WRD
heavy accents	ChangSha	31	9000	75698
	JiNan	31	9000	75674
	LanZhou	30	9000	75653
	NanJing	31	8999	75680
	TangShan	31	8999	75638
	XiAn	31	8999	75662
	ZhengZhou	31	9012	75799
light accents	Beijing	33	9140	76987
	ChangChun	32	9298	78340
	ChengDu	31	9000	75672
	FuZhou	31	8999	75656
	GuangZhou	32	9279	77777
	HangZhou	31	8999	75674
	NanChang	31	8998	75650
	ShangHai	31	9002	75669

standard Mandarin, while the deviation from the standard Mandarin of the remaining seven accents is only marginal. Therefore, for most of the evaluations in this paper, we divide the complete data set into two categories, light vs. heavy accents, as shown in Table 6.3.

6.4.2 Baseline

The baseline AM has more than 20 M weights and is trained on several thousands of hours of recordings related to the in-car domain. The acoustic features comprise 45 Mel-frequency Cepstral Coefficients (MFCCs) and seven fundamental frequency variation (FFV) features [154], extracted at a frame rate of 10 ms and a window size of 25 ms. The FFV features are added to capture Mandarin tones. The input feature vectors have a context of 15 frames (7-1-7) and they are finally concatenated with a 100 dimensional *i*-vector [155, 165] estimated for 50,000 speaker clusters. The output layer of the DNN contains 9,000 nodes, corresponding to context-dependent phone HMM states. The phone set consists of 55 consonants and 21 vowels. All vowels have separate models for the five tones (the only exception is that 'ER' does not occur with the first tone). Thus, the phone set contains $(55 + 21 * 5 - 1) = 159$ phones.

The LM is composed of a word-based 4-gram model and a recurrent neural network (RNN) model. The RNN has two recurrent hidden layers, both containing 1000 nodes. The vocabulary size is beyond 400k. A mixture of field data and grammar-based artificial data is used for training. We used the same LM in all experiments.

6.4.3 Toneless and Sixth-Tone AM

The toneless phone set (see Section 6.2.2) collapses all tonal versions of vowels into one toneless version. This modification is applied directly to the original tonal segmentations – simply by removing the tone information from all vowels. The sixth-tone phone set (see Section 6.2.3) adds the toneless vowels to the original phone set, resulting in a set of $(159 + 21) = 180$ phones. When re-training the new sixth-tone AM, the full training data was split into two portions. The first portion kept the original transcription with 159 phones (with five tones), and the second portion was used with the new toneless transcription. Five additional AMs were trained, by varying the size of the two portions: 90%/10%, 75%/25%, 67%/33%, 50%.50% and 0%/100% (toneless model).[1] In this way, the toneless AM becomes a special case in the sixth-tone family. The toneless AM was used in conjunction with a lexicon in which all vowels were toneless. For example, the vowels AA1, AA2, AA3, AA4 and AA5 were all replaced by AA.

The effects differ very much between accents. However, the trends are as expected: improvement from left to right for the heavy accents, but deterioration for the light accents. All new AMs were trained independently with the same training configuration of the neural network size and total amount of training data as the baseline. The CERs obtained with the baseline AM and the five new AMs are shown in Table 6.4. It is interesting to point out that the performance for the TangShan accent is better than for many accents classified as 'light'. Apparently, human judgments about accent strength do not always correspond to problems encountered by ASR systems.

[1]The first experiment was done using half of the training data with toneless transcriptions/pinyins. The results for the light accents were disappointing. For that reason, we decided not to test AMs with even larger proportions of toneless transcriptions.

TABLE 6.4: *CER for the baseline and sixth-tone acoustic models. The headings .9.1 etc. indicate the proportion of the training data allotted to original segmentations and toneless segmentations, respectively.*

	system / testset	base	.9.1	.75.25	.66.37	.5.5	toneless
heavy accents	ChangSha	11.23	10.83	10.55	10.07	10.43	10.63
	JiNan	11.84	10.72	10.03	10.68	10.03	8.45
	LanZhou	19.67	17.08	16.67	15.29	14.44	13.37
	NanJing	12.96	12.78	12.13	12.48	12.47	11.51
	TangShan	8.00	7.88	7.80	7.70	7.79	7.65
	XiAn	15.45	14.07	13.29	13.13	12.1	10.14
	ZhengZhou	17.01	14.61	13.47	13.24	11.81	9.57
	avg. heavy	13.71	12.54	11.97	11.78	11.28	10.16
light accents	Beijing	7.15	7.15	7.51	8.25	8.16	9.20
	ChangChun	4.80	4.94	5.27	5.46	5.81	6.12
	ChengDu	10.68	11.17	12.21	11.67	12.43	13.41
	FuZhou	6.57	7.15	6.50	7.42	7.26	8.73
	GuangZhou	11.93	12.14	12.64	14.12	13.92	14.09
	HangZhou	10.11	10.20	10.05	10.80	10.84	11.1
	NanChang	9.01	9.53	9.94	10.06	10.76	10.51
	ShangHai	8.18	8.94	9.27	10.04	9.76	9.86
	avg.light	8.49	8.83	9.09	9.65	9.78	10.31

6.4.4 Alternative Approach: Full Tonal Extension

Instead of the expensive AM training, the effect of the sixth tone can be approximated by adding all the tonal alternatives to the representations in the lexicon. For example, the word "马" corresponds to a phone sequence 'M AA3'. We then add the other major tones of 'A' ('M AA1', 'M A2' and 'M AA4') as alternative pronunciations of "马".[2] Our original Mandarin vocabulary comprises 1810 syllables. By adding three pronunciation variants for each syllable, that number increases to 5430. In the case of compound words with multiple characters, which are very common in Mandarin, we modified the lexicon in a cascaded way: we first add tonal variants for the first character while keeping all other characters fixed. Then, the pronunciation of the second character is changed with the remaining characters unchanged from the original pronunciation, including the first character and so on.

[2]Note that we did not add the 'weak' tone number 5 as an alternative pronunciation, because this would surely result in spurious confusions.

TABLE 6.5: *Comparison of CERs of sixth-tone model and the lexicon alternative across all accents.*

	system / testset	toneless	lex_all
heavy accents	ChangSha	10.63	10.53
	JiNan	8.45	8.86
	LanZhou	13.37	14.78
	NanJing	11.51	11.38
	TangShan	7.65	7.86
	XiAn	10.14	10.29
	ZhengZhou	9.57	10.64
	avg. heavy	10.16	10.59
light accents	Beijing	9.20	9.12
	ChangChun	6.12	6.56
	ChengDu	13.41	13.12
	FuZhou	8.73	8.86
	GuangZhou	14.09	13.76
	HangZhou	11.10	11.57
	NanChang	10.51	11.01
	ShangHai	9.86	10.09
	avg.light	10.31	10.44

We compared the CERs with the toneless AM from Section 6.4.3 with the CERs obtained with the expanded pronunciation variants in the lexicon. The results of the comparison are shown in Table 6.5. The marginal differences between the two systems across all 15 accents show that the sixth-tone model can be simulated by adding 'all confusions' in the lexicon. Both approaches are conceptually close. A Wilcoxon Matched Pair test showed that the difference between the performance for the 'all confusions' and 'sixth tone' approaches is not significant.

6.4.5 Enumerate Tonal Confusions

As described in Section 6.3.2, to obtain more fine-grained data about tonal confusions, we counted these confusion when using the baseline AM and the baseline lexicon. Since the data in Table 6.4 show that adding tonal confusions only improves CERs for the heavy accents, this part of the experiments was limited to the heavy accents. We created 12 lexicons, each of which reflects one of the twelve possible tone confusions. With these lexicons we investigated the impact of individual

tone confusions in the seven accents in the development set. The results of these experiments are shown in Figure 6.2. In each panel the five confusions that have the largest impact are underlined in red. With the exception of NanJing and TangShan the majority of the individual confusions decrease the CER. However, it is not guaranteed that combining all confusions that individually have a positive impact in a specific accent yield optimal CERs. We experimented with lexicons that included the top $N, N = 1, 2, \ldots, 12$ confusions. Eventually, accent-specific lexicons are created that contain the five most important confusions for each accent. With those lexicons we performed a recognition experiment on the test sets for all 15 accents. From Figure 6.2 it can be inferred that the top five confusions for JiNan and ZhengZhou are the same, be it in a different order.The same holds for another group, comprising ChangSha, NanJing and TangShan. As a result, the accent-specific lexicons for these groups of accents are identical, leaving us with only four different accent-specific lexicons.

The results of the recognition experiments with the accent-specific lexicons are summarized in Table 6.6. In the data for the heavy accents the best-performing accent-accent combinations are shown in bold. The right-most column contains the CERs that were obtained with the baseline lexicon (copied from Table 6.4). The table shows that accent-specific lexicons improve the performance for all seven heavy accents. From the lower part of the table it appears that all light accents suffer from adding tone confusions to the lexicon: all accent-specific CERs are higher than the CER in the baseline column. It is also evident that with the exception of TangShan all heavy accents show the best performance when the accent-specific lexicon is used. Actually, from Figure 6.2 one would have expected a slight negative effect from adding the top-five confusions for NanJing. However, the development and test sets contain only five speakers. We attribute the positive effect of the top-five confusions for NanJing and the negative effect for TangShan to mismatches between development and test sets. The risk that the development sets might not be fully representative was an important argument for the decision to take the top-five confusions for all accents, rather than the N_{accent} that yielded the best result for an accent.

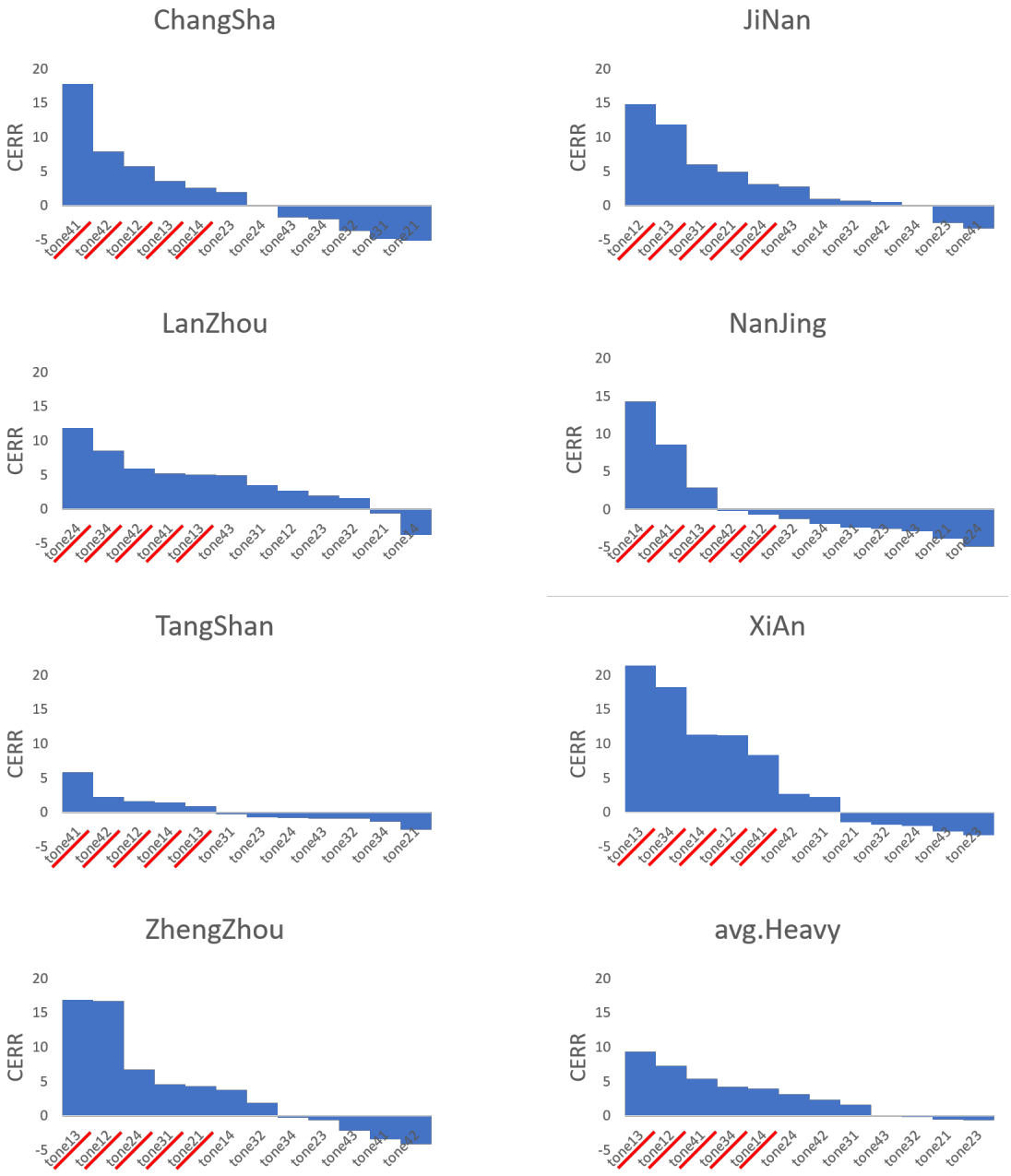


FIGURE 6.2: CERRs based on individual tonal confusions, aiming to find the confusions that contribute the most per accent in the dev set. The top 5 tonal confusions are underlined.

TABLE 6.6: *CER of tonal level confusions on test set*

	system / testset	ChangSha	JiNan	LanZhou	NanJing	TangShan	XiAn	ZhengZhou	baseline
heavy accents	ChangSha	**9.73**	11.84	9.71	9.73	9.73	9.82	11.84	11.23
	JiNan	10.01	**9.36**	9.99	10.01	10.01	10.15	9.36	11.84
	LanZhou	16.96	16.40	**13.95**	16.96	16.96	18.45	16.40	19.67
	NanJing	10.82	12.51	11.67	**10.82**	10.82	10.75	12.51	12.96
	TangShan	7.73	**7.19**	7.82	7.73	7.73	7.72	**7.19**	8.0
	XiAn	11.06	12.81	10.01	11.06	11.06	**9.99**	12.81	15.45
	ZhengZhou	13.64	10.78	12.64	13.64	13.64	13.2	**10.78**	17.01
	avg. heavy	11.39	11.52	10.81	11.39	11.39	11.4	11.52	13.71
light accents	Beijing	8.42	8.67	8.61	8.42	8.42	8.38	8.67	7.15
	ChangChun	5.72	5.54	5.83	5.72	5.72	5.68	5.54	4.8
	ChengDu	12.58	12.32	12.88	12.58	12.58	12.56	12.32	10.68
	FuZhou	8.70	8.02	8.71	8.70	8.70	8.37	8.02	6.57
	GuangZhou	14.28	14.05	14.67	14.28	14.28	14.49	14.05	11.93
	HangZhou	11.78	11.41	11.66	11.78	11.78	11.56	11.41	10.11
	NanChang	10.33	10.31	10.53	10.33	10.33	10.22	10.31	9.01
	ShangHai	10.44	10.26	10.62	10.44	10.44	10.37	10.26	8.18
	avg.light	10.21	10.00	10.36	10.21	10.21	10.13	10.00	8.49

6.4.6 Typical Consonant Confusions

As described in Section 6.3.3, a list of consonant confusion pairs are evaluated based on phonetic knowledge about Mandarin. The list can be found in Table 6.2. The approach is the same as for tone confusions in Section 6.4.5: we created 14 different lexicons, each reflecting a specific consonant confusion. The resulting CER changes obtained with the development test set are shown in Table 6.7. It can be seen that the added consonant confusions mainly cause marginal degradations. Despite the fact that consonant confusions do occur frequently, Table 6.7 shows that adding pronunciation variants based on consonant confusions do not improve recognition accuracy. Adding accent-specific consonant confusions does not improve accuracy either. A detailed analysis of the impact of consonant confusions showed a substantial difference between the speakers: for some speakers we saw substantial improvement, but for others equally substantial degradation.

6.4.7 Syllable-level Solutions

Mandarin Chinese has 1,810 different syllables, for which pronunciation variants can be constructed in multiple ways. By adding pronunciation variants to the entries in the lexicon as described in Section 6.3.4 and keeping track of the variants selected by the decoder, we are able to determine which variants improve recognition accuracy for a specific accent. For example, the tone confusion $C34$ adds the variant from MAA4 to MAA3, but also from BEI4 to BEI3. However, it is possible that in a certain accent only $MAA3 \rightarrow MAA4$ improves the CER, while $BEI3 \rightarrow BEI4$ contributes nothing; it might even deteriorate CER. Having said this, there are still many ways in which pronunciation variants can be selected for inclusion in a lexicon. We used four selection strategies, starting from (1) the top-five tone confusions (syl(top5)), (2) all consonant confusions (syl(con)), (3) all twelve tone confusions (syl(ton)), and (4) the combination of all tone and all consonant confusions (syl(all)). For each of the seven heavy accents, the syllable variants that improved CER on the development set per accent compose the confusion sets, which are then added to accent-specific lexicons. The results on test sets are shown in the leftmost part of Table 6.8. The CERs for the light accents in the lower half of that Table are the mean of the accuracies obtained with the seven accent-specific lexicons.

TABLE 6.7: *Changes in CER per accent for 14 consonant confusions.*

	system / devset	ag2an	eg2en	ig2in	c2ch	ch2c	s2sh	sh2s	z2zh	zh2z	l2n	n2l	f2h	q2x	x2s
heavy accents	ChangSha	-2.87	0.43	-0.36	0.07	-0.86	-0.36	-2.15	-0.36	0.00	-0.14	-0.50	0.22	-0.57	-1.29
	JiNan	-3.05	0.00	-0.58	-0.14	-0.51	-0.73	-1.67	-0.29	-0.22	-0.73	-2.25	-0.73	-0.94	-1.09
	LanZhou	-2.34	0.28	0.31	-0.42	-0.07	-0.35	-1.08	0.28	-0.52	-0.38	-1.29	0.07	-0.35	0.00
	NanJing	-1.83	-0.06	0.18	-0.06	0.00	0.00	-0.18	-0.24	0.06	-0.73	-1.47	-0.49	0.06	-0.86
	TangShan	-2.74	-1.06	-1.15	-0.97	0.09	-1.59	-0.80	-0.80	-0.18	-0.44	-1.15	-0.09	-2.3	0.35
	XiAn	-3.31	-0.58	-0.42	-0.74	-0.26	-0.58	-1.79	-0.89	-0.74	-0.58	-1.16	-0.37	-1.05	-1.42
	ZhengZhou	-2.67	0.16	-0.27	-0.55	-0.87	0.87	-1.31	0.22	-0.38	-0.60	-1.15	-0.44	-1.25	-2.51
	avg.heavy	-2.65	-0.06	-0.22	-0.40	-0.34	-0.31	-1.27	-0.22	-0.33	-0.51	-1.28	-0.24	-0.82	-0.96
light accents	Beijing	-2.20	-1.00	-0.60	-2.20	0.00	-2.39	-1.40	0.00	0.00	-0.40	-1.40	-0.40	-0.6	-1.60
	ChangChun	-4.27	-0.32	-0.63	-1.58	0.16	-0.63	-1.26	-0.95	0.00	-0.32	-2.05	-0.32	-1.9	-0.63
	ChengDu	-1.64	-0.33	-1.20	-0.11	0.00	0.00	0.33	0.00	0.98	0.55	0.87	-0.44	-0.77	-1.31
	FuZhou	-0.38	0.39	-0.90	-0.51	-0.64	-0.77	-1.93	0.39	-0.38	-1.03	-2.31	-0.26	-2.18	-0.38
	GuangZhou	-0.11	0.00	-1.68	-0.11	0.00	-0.78	-1.12	0.00	-0.45	-0.45	-1.34	-0.11	-1.01	0.22
	HangZhou	-0.32	0.00	-0.74	0.00	-0.11	-0.53	2.01	-0.42	1.06	-0.85	-1.27	-0.42	-2.75	-1.06
	NanChang	-1.06	-0.18	2.13	0.00	-0.09	0.53	0.18	-0.44	-0.80	-0.80	-0.62	0.00	-1.77	-1.15
	ShangHai	-1.68	0.00	0.11	-0.45	0.00	-0.22	0.00	-0.22	0.56	-0.11	-2.13	-0.34	-0.56	-1.23
	avg.light	-1.30	-0.13	-0.33	-0.46	-0.09	-0.45	-0.24	-0.21	0.12	-0.43	-1.20	-0.27	-1.48	-0.88

From the baseline CERs (rightmost column) it can be seen that variants that only involve consonants (the column 'syl(con)') have a very small effect at best. This is in line with the results of the experiment described in Section 6.4.6. The fact that consonant confusions do not help also explains the observation that the columns 'syl(ton)' and 'syl(all)' are very similar.

For the seven heavy accents the column 'tone(top5)' in Table 6.8 contains the numbers on the main diagonal in Table 6.6. The fact that the numbers in the column 'syl(top5)' are not always smaller than the corresponding numbers in the column 'tone(top5)' shows that the laborious procedure for selecting the syllables that improve accuracy is not foolproof. The difference is especially apparent for the LanZhou and ZhengZhou accents, which have the highest CERs.[3]

From the numbers in the left-most four columns in the lower part of Table 6.8 it can be seen that the CERs for all light accents are essentially equal to the baseline. Remember, however, that these numbers are averages of the accuracies obtained with the accent-specific systems, which do not exist in an operational ASR system.

6.4.8 Accent-independent Solution

All experiments thus far aimed at constructing lexicons that optimized CER separately for each of the seven heavy accents. Now, we aim to develop one lexicon that maximizes performance for all seven heavy accents at once. Because the CERs for the light accents, when averaged over the CERs obtained with lexicons optimized for each of the heavy accents, are essentially equal to the results obtained with the baseline lexicon, we expect that such a lexicon will also be acceptable for the light accents. We used a procedure similar to the one employed in building optimal lexicons for individual accents, but the experiment was limited to two pools of potential variants, viz. the union of the variants in the lexicons of the heavy accents in 'syl(ton)' and 'tone(top5)' in Table 6.8. Because a full search to find the globally optimal variants is prohibitive, we used a couple of heuristics, such as a lower bound for the number of utterances in which a variant improved recognition and an upper bound for the number of times that a variant caused errors.

[3] $CER = 10.04$ in the column 'syl(all)' is within the 5% confidence interval around 10.78 in 'tone(top5)' with 12,000 observations. The same holds for the 14.21 in 'syl(all)' relative to the 13.95 in 'tone(top5)' for LanZhou'.

TABLE 6.8: *CER of Accent-independent syllable level confusions*

	accent-specific				accent-independent			
	syl(top5)	syl(con)	syl(ton)	syl(all)	syl(glob)	tone(glob)	tone(top5)	base
ChangSha	**9.07**	11.14	**9.07**	9.09	9.40	10.86	9.73	11.23
JiNan	10.09	11.84	9.56	9.56	9.14	**8.99**	9.36	11.84
LanZhou	15.08	19.63	14.19	14.21	14.62	**13.63**	13.95	19.67
NanJing	10.94	12.95	10.61	10.64	**10.59**	11.88	10.82	12.96
TangShan	7.25	7.98	7.01	7.01	**7.01**	7.26	7.73	8.00
XiAn	**9.45**	15.48	9.47	9.50	10.12	12.26	9.99	15.45
ZhengZhou	11.69	17.00	11.04	11.04	11.96	11.51	**10.78**	17.01
avg.heavy	10.51	13.72	**10.14**	10.15	10.38	10.89	10.32	13.71
	syl(top5)	syl(con)	syl(ton)	syl(all)	syl(glob)	tone(glob)	tone(top5)	base
Beijing	7.31	7.17	7.33	7.33	7.20	8.41	8.51	**7.15**
ChangChun	4.93	**4.80**	5.01	5.02	4.85	5.43	5.68	**4.80**
ChengDu	10.98	**10.63**	11.06	11.06	10.65	11.85	12.55	10.68
FuZhou	6.79	6.58	6.84	6.83	**6.57**	8.36	8.46	6.57
GuangZhou	11.93	11.86	11.96	11.97	**11.64**	13.92	14.30	11.93
HangZhou	10.08	10.07	9.93	9.92	**9.55**	11.52	11.63	10.11
NanChang	9.16	**9.01**	9.25	9.25	9.15	10.08	10.34	9.01
ShangHai	8.31	8.17	8.38	8.38	**8.19**	10.32	10.40	8.18
avg.light	8.62	8.47	8.65	8.65	**8.41**	9.92	10.16	8.49

The results of this experiment are shown in the columns 'syl(glob)' and 'tone(glob)' in Table 6.8. It can be seen that -on average- the CERs for the heavy accents are very similar with the two lexicons. The only outlier is LanZhou, the accent with the worst baseline performance. Apparently, at least part of the syllable variants that are needed for LanZhou are left out because they have a negative effect on the other six heavy accents. This is supported by Figure 6.2 that shows that for LanZhou ten out of twelve tone confusions improve accuracy when implemented individually. It can also be seen that 'syl(glob)' does not really improve the performance for the light accents, but that it does not degrade the performance either. Thus, the 'syl(glob)' achieved one reasonable lexicon for all 15 accents.

6.4.9 The Impact of Speaker Variation

In Sections 6.4.5 and 6.4.6 we investigated how the 12 individual tone and 14 individual consonant confusions affect the CER in the 15 accents, with CERs per accent averaged over the five speakers in the test sets. The results suggested that accounting for tone confusions was advantageous for the seven heavy accents, while the consonant confusions never appeared to improve performance. Because the number of speakers in each of the 15 test sets is so small, we decided to investigate

whether the overall results might be affected by differences between speakers. For this purpose we use heat-map plots of the proportional change of CER for all individual speakers. To avoid large changes to obscure small ones the color coding is constrained to $(-10\%, 10\%)$.

Figure 6.3 shows the heat-map plots for the speakers of the Beijing and ChangChun accents. Because both are standard Mandarin, one would expect that the default lexicon is fully adequate, and that adding confusions might cause detrimental confusability. This is confirmed by the plots. If anything, the effect of tone confusions is worse than that of consonant confusions.

The heap-map plots in Figure 6.4 show the speaker-specific impact of the confusions in the seven heavy accents. The gains from the tone confusions are are evident, the more so if it is realized that the color coding is limited to $(-10, 10)$. In some cases improvement can be as high as 38%. The plots confirm that the tone confusions have a much bigger impact than the consonant confusions. It can also be seen that spk5 in both JiNan and LanZhou differ substantially from the other four speakers of those accents.

The heat-map plots for the remaining six light accents in Figure 6.5 confirm that there too adding tone confusions to the lexicon mostly deteriorates CERs. However, there are some red blocks, for instance, C23 for SPK5 in FuZhou, C23 for SPK3 in GuangZhou and C14 for SPK3 in HangZhou. This illustrates that also some speakers from light accent regions might benefit from adding pronunciation variants to the lexicon. The light accents deviate from standard speech at some extent on the tonal side. Here, the relative effects of consonant confusions is larger, but it is not possible to find specific confusions that are expected to yield improvements for most speakers in all accents.

6.4.10 Trade-off between Light/Heavy Accents and CER/RTF

Adding pronunciation variants to a lexicon inevitably increases the number of confusable pronunciation; as a consequence, the search will become more complex and require more computational resources. The increase in computational resources is a function of the number of added variants, but because not all variants make equal contributions to confusability, that function may not be linear. Therefore, we conducted an experiment to uncover the trade-off between the number of

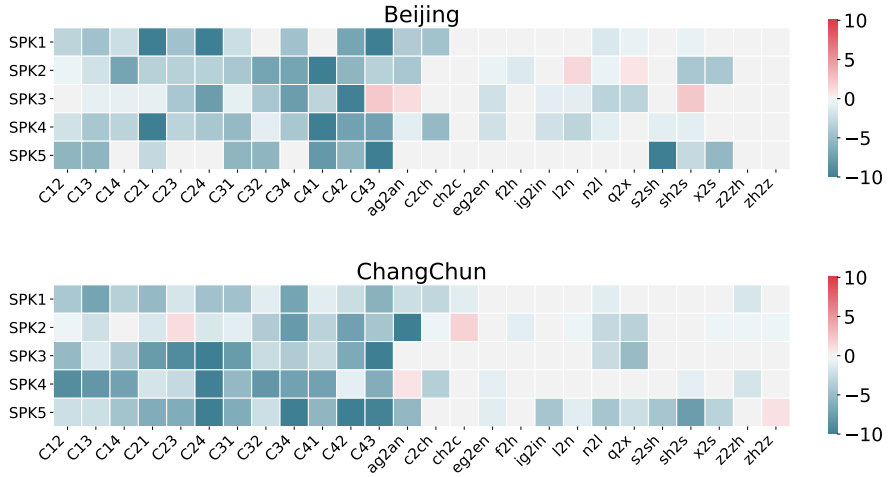


FIGURE 6.3: Heat-map plot of CERRs based on individual tonal or consonant confusions per speaker for standard Mandarin.

variants and the real-time factor in our ASR system. For this experiment we added pronunciation variants to the $10, 100, 1000, 10,000, 100,000$ most frequent words in our 400k lexicon. We did this with the pronunciation variants selected in the 'syl(glob)' and 'tone(glob)' lexicons.

Both CER and RTF are averaged over the heavy and light accents, for each of the two lexicons.

From Figure 6.6 it can be seen that adding pronunciation variants to ever more entries in the 'syl(glob)' lexicon has no effect on the CER for the light accents. However, adding variants to more words in the 'tone(glob)' lexicon increases the proportion of errors. As could be expected from the numbers in Table 6.8 the effect for the heavy accents is the opposite: adding variants improves CER, and the effect starts out stronger for the 'tone(glob)' lexicon. The latter effect reflects the LanZhou effect observed above. The RTF as a function of the logarithm of the number of words with variants is approximately linear in the 'tone(glob)' lexicon; with the 'syl(glob)' lexicon the relation between RTF and number of words with variants shows a bent when going from 1,000 to 10,000 entries.[4] The RTF data are -by necessity- approximate, because measurements were made in an operational service.

[4]In an operational system an increase of the RTF of 2-3% is often deemed acceptable.

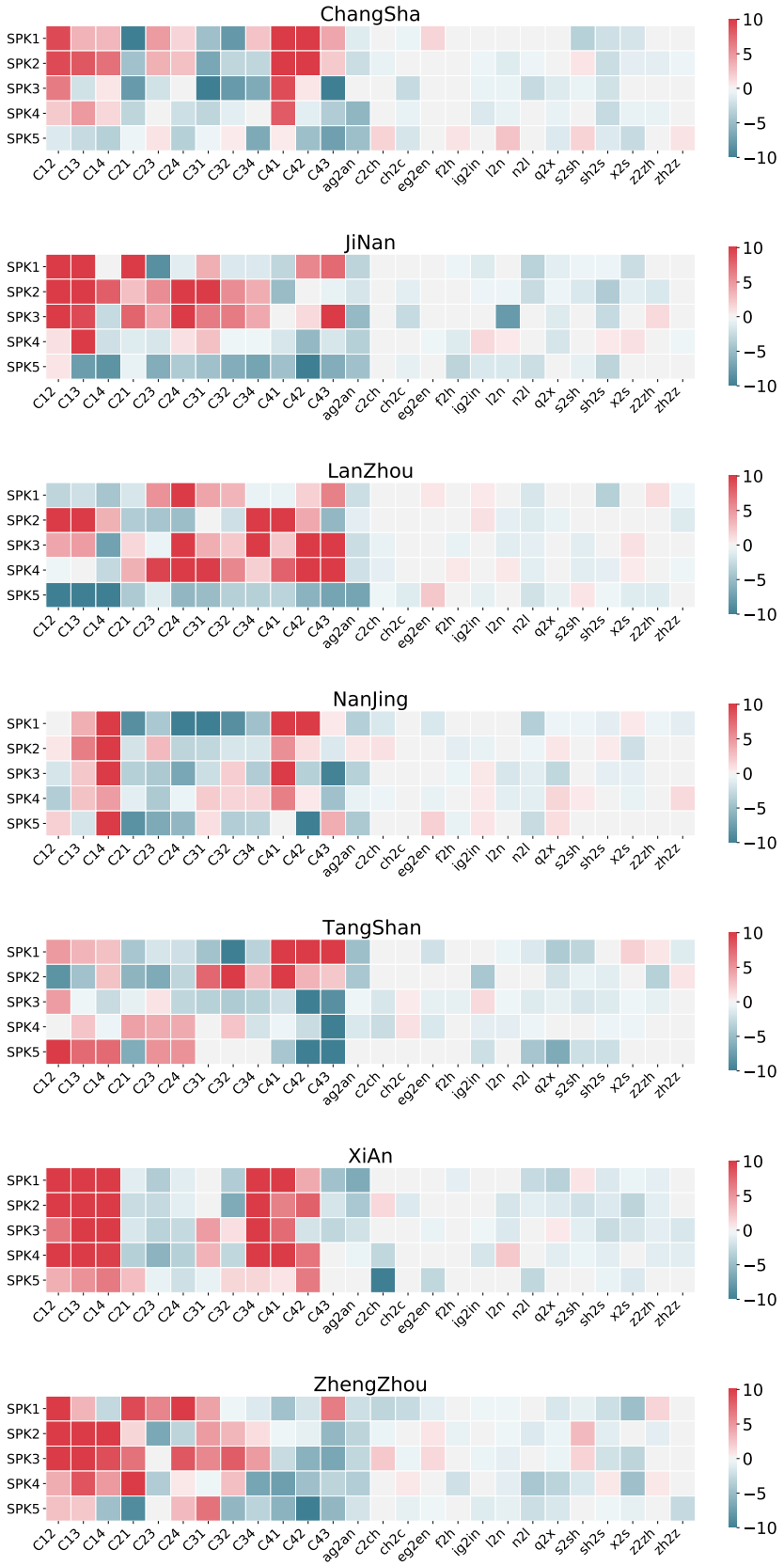


FIGURE 6.4: Heat-map plot of CERRs based on individual tonal or consonant confusions per speaker for heavy accents.

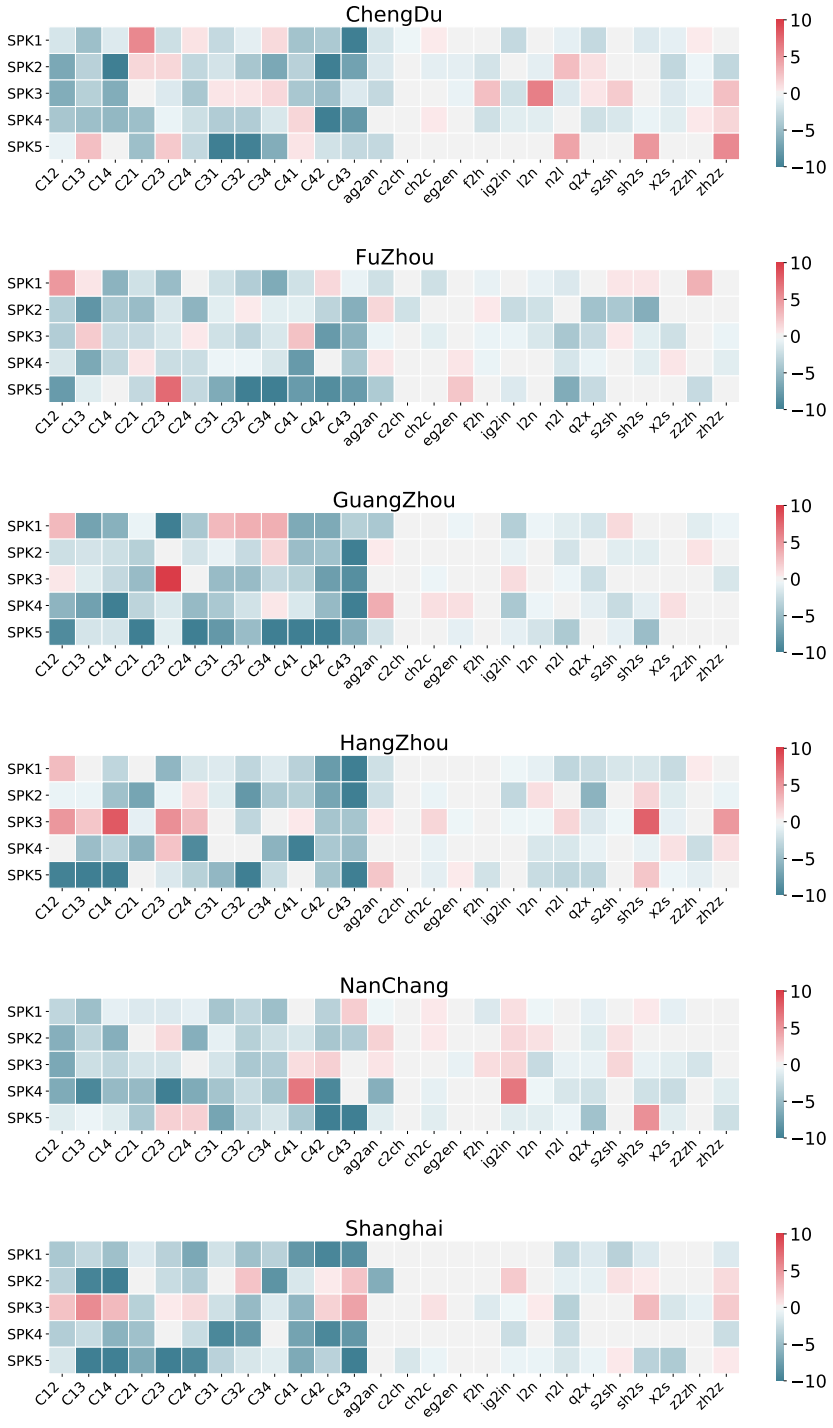


FIGURE 6.5: Heat-map plot of CERRs based on individual tonal or consonant confusions per speaker for light accents.

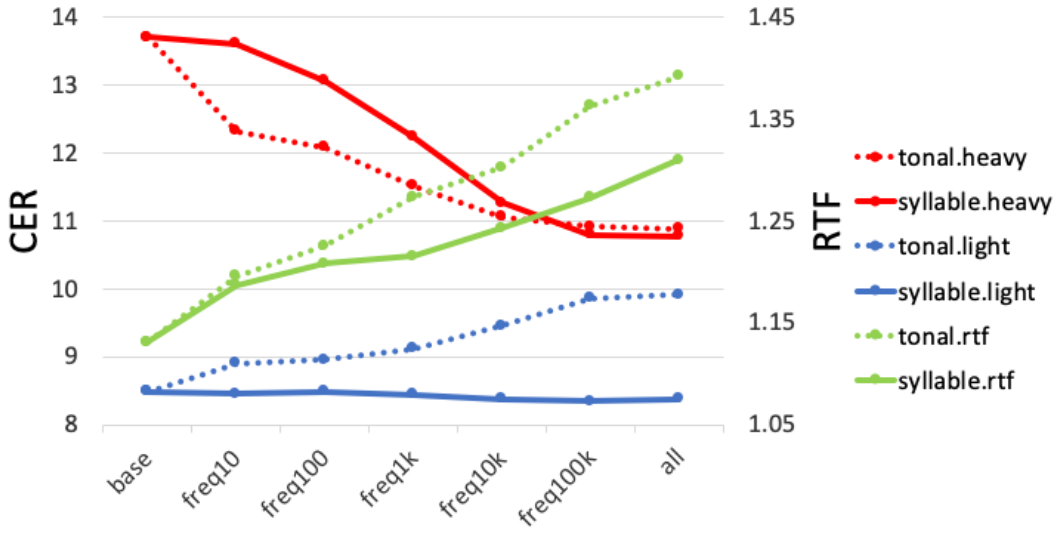


FIGURE 6.6: Trade-off between CERs of light and heavy accent and between CER and RTF.

6.4.11 Fully Data-driven Approach

While the experiments described in sections 6.4.5 and 6.4.6 do involve analyzes of data about confusions and CERs, the design of those experiments is completely guided by phonetic knowledge. This introduces the risk of missing confusions that have not been considered by phoneticians. For this reason we decided to embark on an approach that makes it possible to discover all confusions that explain recognition errors and that might be counteracted in ways that do not complicate the decoder. One way for doing this is by comparing the output of an unconstrained phone classifier with the classification based on a forced alignment with the canonical phone transcription of the sentences.

The 9,000 nodes in the output layer of the DNN represent the same number of clustered triphone states. We mapped the triphones to the 159 mono-phones by taking the label of the middle phone. By doing so we can construct a frame-based matrix of confusions between all 159 mono-phones. From this 159×159 matrix we can distill numerous smaller matrices that focus on specific confusions. To test the feasibility of this approach we decided to compare frame-based confusions with the phone-based confusions between the four major tones in Section 6.4.5. As done in the experiments described above, we counted frame-based confusions using the development set. The results (again only for the heavy accents) are summarized in Figure 6.7.

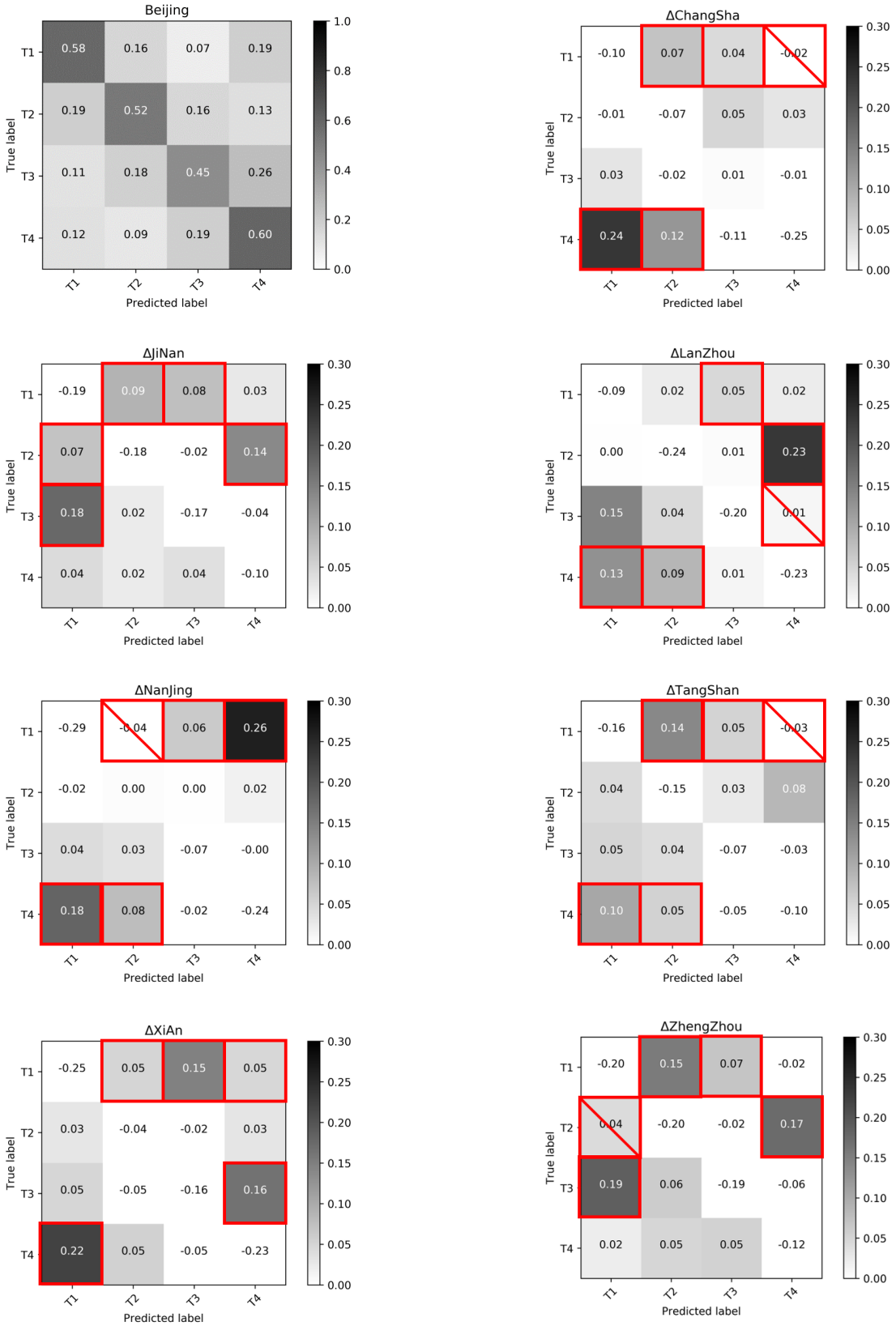


FIGURE 6.7: Data-driven approach to find tonal confusions

The panel (top left) showing the confusions for the Beijing accent is different from the panels for the seven heavy accents. The Beijing accent is considered as 'standard Mandarin' and the corresponding panel is the only one showing actual proportions of confusions, represented such that the proportions in each row sum to one. It is obvious that the frame-based classification is far from perfect, even for the standard accent. Mis-classifications can have several causes, such as Sandhi tone [174, 175], and differences in the segmentations between the forced alignment and the unconstrained decoding. It appeared that the raw confusion matrices for the seven heavy accents were difficult to interpret. However, the difference between raw matrices and the Beijing matrix, displayed in the remaining seven panels in Figure 6.7 do provide interesting insights.

The five confusions that appeared to affect recognition performance most in Section 6.4.5 are shown with red borders in the panels for the individual accents. When the corresponding frame-based confusion $\Delta(CM_{accent} - CM_{Beijing})$ does not rank in the top five, it is marked by a red main diagonal. It can be seen that overall there is a good correspondence between the phone-based and frame-based approaches, but the correspondence is not perfect. From the numbers on the main diagonal in the Beijing matrix it can be seen that tone #3 causes more problems than the three other tones. The numbers on the main diagonal in the matrices for the seven heavy accents show substantial differences between the accuracy for the tones in Beijing and the individual accents. For example, in NanJing the difference with Beijing for tone #2 is negligible. But the -0.24 in the T4-T4 cell shows that the accuracy for NanJing is substantially worse than in Beijing. This that was not visible in the phone-based approach in Section 6.4.5.

It is worth pointing out that the five cases where there was no overlap between the knowledge-driven and data-driven top-five correspond to a relatively large off-diagonal entry in the 'Beijing' matrix. A clear example is the confusion 'C34' in the 'LanZhou' matrix. The proportion of tone #3 tokens in LanZhou that are recognized as tone #4 is 0.27; subtracting the proportion of 'C34' confusions in Beijing (0.26) we are left with a difference between the confusions in the two accents of 0.1, which is not in the top-five differences. While some form of normalization with respect to the confusions in the 'standard' accent is necessary, normalization by means of simple subtraction is not always optimal for the purpose of identifying the top-N confusions that should be used for adding pronunciation variants to the lexicon.

6.5 Discussion and Outlook

6.5.1 AM training vs. Lexicon Modification

It is well known that Mandarin accents differ between each other by using different tones without changing the vowels. To tackle that problem for all accents simultaneously we introduced toneless phones (see Section 6.2.2). This can be done by training new acoustic models, or by adapting the lexicon. In Section 6.4.4 (see Table 6.5) it is shown that for all 15 accent CERs obtained by adding toneless vowel models to the AM or adding all four major tones to each vowel in the lexicon are essentially identical. Subsequent sections showed that additional improvements can be obtained by adding accent-specific pronunciation variants to the lexicon, instead of simply adding all tones to each vowel.

Our AM training approach does not attempt to explore accent-specific problems. Instead, it aims to provide a general solution for all Mandarin accents. We do not attempt to train accent-specific AMs, because this would need substantial amount of training data consisting of various speakers per accent. In addition, training a new AM is much more expensive than making changes to the lexicon. Training a full AM can take up to a week, while changing the lexicon is almost instantaneous. Another advantage of the lexicon approach is that a relatively small data set suffices to identify accent-specific confusion patterns that can then be used to solve problems.

The lexicon approach is also advantageous in actual deployment of an ASR system. One can imagine scenarios such as code-switching system or system combination to optimize accuracy, because deploying lexicons in parallel is much cheaper than parallel AMs. It also allows nightly updates of the lexicon to patch the ASR system for specific customer use cases. However, lexicon modification is a hard-switching approach. Either an accent classifier is required or a trade-off needs to be made between different accents and also between CER and RTF as shown in Figure 6.6. Of course, the same holds for accent-specific AMs.

6.5.2 The Impact of Speaker Variation

It is well known that creating speaker-independent ASR systems is extremely difficult, because of the speaker variation in gender, age, articulation rate and multiple other factors. The situation becomes much worse when it comes to accented speech, which is also affected by cultural factors, such as distance from mother tongue to standard speech, education, regional history, etc. Effects of accent-internal speaker variations are evident in Figures 6.4 and 6.5. For example, most tone confusions provide good gains for 'SPK 1-4' in the JiNan and LanZhou accents. However, 'SPK5' in both accents reveal a very different situation. A similar effect is found in GuangZhou accent, where the tone confusion rule 'C23' provides over 9.85% relative CERR for 'SPK3', whereas the same tonal confusion degrades 'SPK1' by -10.15%.

Speaker variation makes it difficult to obtain representative development sets. In this chapter, we split the data sets for each accent, which comprise around 30 speakers, by the ratio of 4-1-1. All the lexicon-based confusion rules are designed on the basis of the five speakers in the dev-set of each accent and evaluated on five other speakers in the test set. Our experiments show that this approach is sub-optimal; the accent-specific confusion rules derived from five speakers do not always yield the best results for the target accent. For example, TangShan's performance of 7.73% on the diagonal line of Table 6.6 is outperformed by the lexicon based on JiNan's rule (7.19%). A leave-one-out cross validation strategy might alleviate this problem, be it at a substantial cost.

6.5.3 Comparison between Two Data-driven Approaches of Lexicon Modification

The tone confusion pairs that are in the top-five in the knowledge-driven approach (see Section 6.4.11) are shown in boxes with red borders in Figure 6.7, which summarizes the proportions of confusions found with the data-driven approach. If a top-five knowledge-driven confusion is not in the data-driven top-five, the corresponding confusion is marked by a red diagonal. For JiNan and XiAn the top-five in the two approaches are the same. In the five other accents there is only one tone confusion in the knowledge-driven approach that is not also in the top-five of the data-driven approach. We compared the CERs with the accent-specific

lexicons in Table 6.6 with accent-specific lexicons based on the top-five confusions found with the data-driven approach. The results are summarized in Table 6.9. Evidently, for JiNan and XiAn the difference is zero, because the top-five are identical. For ChangSha, NanJing and TangShan the data-driven top-five yields small CER improvement. However, for LanZhou and ZhengZhou the knowledge-driven approach yields slightly lower CERs. The LanZhou accent misses the 'C34' confusion in the data-driven top-five. In Section 6.4.11 the cause of this miss is explained, despite the fact that the 'C34' confusion is important in LanZhou.

The idea of the knowledge-based approach is to try out all confusions and see which ones improve the performance most. It is almost impossible to test all possible phonetic confusions, so that linguistic knowledge is required to pre-filter the most promising ones, such as the tone and consonant confusions investigated in Section 6.4.5 and 6.4.6, respectively. The fully data-driven approach, on the other hand, aims at finding mismatches between the ground truth reference and phone sequences predicted by the acoustic model at the phone level, where the alignment only needs to be done once to display the complete view of possible accent-specific phonetic confusions. Then, selected confusions can be applied directly in recognition tests. It is a promising result that Figure 6.7 shows that the outcome of these two approaches are in line with each other in most accents. This opens the way to investigate other confusion sub-matrices that can be derived from the big 159×159 matrix of phone confusions. Also, the data-driven approach can be applied to languages for which no extensive phonetic knowledge is available.

A shortcoming of both approaches is that the phonetic confusions are evaluated independently. There is no guarantee that the most promising individual confusions will lead to an optimal solution when multiple confusions are combined and jointly applied on the lexicon. It is probably possible to find better top-five sets of tone or syllable confusions by testing multiple confusions at the same time, again at the cost of substantially complicating the experiments.

6.5.4 An Accent Classifier Required?

From Table 6.6 it can be seen that, with the exception of TangShan, accent-specific lexicons that include pronunciation variants that cover accent-specific tone

TABLE 6.9: *Comparison of CERs between enumeration and data-driven approach. The accent-specific top-five tone confusions are spotted by both methods and the CERR is calculated on the development set.*

	enumerate	data-driven	CERR
ChangSha	9.73	9.56	1.75%
JiNan	9.36	9.36	0.00%
LanZhou	13.95	14.00	-0.36%
NanJing	10.82	10.44	3.51%
TangShan	7.73	7.44	3.75%
XiAn	9.99	9.99	0.00%
ZhengZhou	10.78	10.98	-1.86%

confusions yield significant reductions of the CERs relative to the baseline CERs.[5] We did not investigate whether accent-specific lexicons could also lower the CERs for the accents that were classified as 'light' by our transcribers, but that still have double-digit baseline CERs (ChengDu, GuangZhou and HangZhou). Still, the data strongly suggest that the overall performance of the system would improve significantly if a reliable on-line accent classifier could be included. The design and performance of such a classifier is the topic of Chapter 7.

6.5.5 Generalization to Accents of Other Languages

Our results confirm that an approach based on identifying frequent pronunciation variants and adding these to the lexicon is an effective way for improving the overall recognition accuracy of operational ASR systems. As argued above, the lexicon-based approach avoids time-consuming and expensive retraining of acoustic models. Also, it offers more flexibility in adapting operational systems. If detailed phonetic knowledge about a language and its accents is available, a knowledge-driven search for accent-specific confusions can be applied. In the absence of such detailed knowledge the fully data-driven approach is likely to yield improvements.

[5] All bold numbers in the main diagonal in the top half of Table 6.6 are well outside the 5% confidence intervals around the baseline numbers.

6.6 Conclusions

In this work, we studied how to improve accent robustness for Mandarin. Two major approaches are investigated. The first one is to introduce toneless phonemes in the lexicon to forge a generative phone for all 5 tones. In this approach, AM re-training is necessary. The second approach is to modify lexicon which can bridge any accented variations to the potentially more standard phoneme sequences. This approach is more flexible and AM does not need to be re-trained. We studied three levels of phonetic confusions, which gradually filter out the unnecessary and mostly harmful syllable confusions. The performance of the fine-grained syllable-level approach improved in terms of both CER and RTF.

In the next chapter we investigate how to build a robust accent classifier that can improve the performance of an operational ASR system. Future research will focus on the fully data-driven approach. Ideally, the complete syllable confusion list is obtained from the data itself. An improved data-driven approach would help to generalize this idea to any other languages or accents.

Chapter 7

Deep Learning and i-vector based Approaches for Mandarin Accent Identification Towards Accent Robust ASR

7.1 Introduction

In Chapter 6 it was shown that an ASR system optimized for a specific accent of Mandarin Chinese improves recognition accuracy significantly over the accuracy obtained with an accent-independent system. Thus, a system that is able to classify the accent used by arbitrary speakers of Mandarin [176, 177] and allows to select the models that are most appropriate for a speaker of that accent might yield substantial improvements in character error rate. In this chapter we present research in that direction.

There are many languages and dialects in China, which are often not mutually intelligible. Although the only official language is Mandarin in both mainland China and Taiwan, recognizable accents exist under the influence of local dialects. The Standard Mandarin is based on the Beijing dialect, and accents are usually distributed regionally. Geographically, northern dialects in China tend to have fewer distinctions than southern ones (cf. Figure 7.1), while many other factors, such as the history and development of cities, as well as social status and education

level of individual speakers, play an important role as well [161]. As accented speech poses a practical challenge to ASR systems, reliable identification of the accent used by speakers of Mandarin has immediate applications in robust ASR. In this chapter we evaluate various types of classifiers for the Mandarin accent identification task, and propose the usage of a bi-directional Long Short-Term Memory (bLSTM) accent classifier to switch automatically between standard and accented Mandarin pronunciation models (PMs), similar to the models investigated in Chapter 6. We will also investigate the impact of accent classification on CER.

The task of accent or dialect identification has generally received less attention than language identification [178]. Foreign accent identification in English was the subject of the INTERSPEECH 2016 Computational Paralinguistics Challenge [179]. The best performing system [180] in the Challenge used an approach based on the i-vector technique [155]. In the same evaluation framework, comparable performance of i-vector and deep learning based systems [181] was found. For the task of classifying US vs. UK accents in English ASR, a bLSTM based system that is integrated with the acoustic model (AM) was proposed [182], but without using i-vectors. In [183] and [176], the authors presented multi-accent approaches to Mandarin ASR, similar to multi-lingual ASR, using a deep neural network with multiple accent-specific output layers and i-vector input. In a similar vein, [177] introduced accent-specific acoustic features for a Deep Neural Network (DNN) AM. In [184] the authors showed that proper feature selection improves accent classification in Flemish Dutch. In [163], the authors evaluated the performance of speaker adaptation for accented Mandarin speech. However, most previous studies assumed that the accent was known in advance, rather than attempt automatic accent classification. In this chapter we investigate the impact of model selection based on automatic accent prediction. In general, while there are some studies on the tasks of Mandarin tone recognition (e.g., [185, 186]) and tone mispronunciation [187], which are related to the problem at hand, the literature regarding automatic classification of Mandarin accents is scarce and is limited to few, distinct accents [188–190].

The novel aspects of this chapter are: (i) the usage of deep learning and i-vector based systems for discriminating a comprehensive set of regional accents in Standard Mandarin; (ii) a method based on non-metric dimensional scaling (NMDS) for error analysis and subsequent grouping of classes to improve robustness of accent

identification; (iii) reducing the ASR error rate for accented Mandarin speech by using an accent classifier for accent PM selection.

7.2 Data Collection

The speech database used in this study contains 135k utterances (84.7 hours) from 466 speakers. The language is Standard Mandarin as spoken in various regions across China. Fifteen collection locations were selected for broad coverage (see Figure 7.1). All speakers are native speakers of the language or dialect spoken in that region, i.e., they speak the local language or dialect as their first language and learned Standard Mandarin later. Speakers are balanced by gender and across accents (30–32 speakers per accent).

The speech recordings originate from a scripted in-car human-machine interaction scenario.[1] For a realistic setting, the recording was done in mid-size cars (various models per region) while driving on city roads and highways (approximately equal distribution of environments). Moreover, the data collection setup ensures that differences between groups are of acoustic-phonetic, not linguistic nature. All utterances are manually transcribed. Each speaker was recorded in a single recording session. Audio equipment was turned off during all recordings and windows were closed. All data considered in this study were recorded with the same close-talking microphone model (Shidu S-43).

7.3 Accent Identification Experiments

7.3.1 Methodology

For accent identification, we investigate i-vectors [178] as a baseline approach, as well as deep learning methods. The purpose of these experiments is to verify whether the performance improvements from deep learning methods justify the increase in computational complexity compared to i-vector. In particular, we assess the usage of bidirectional LSTM networks [191] to capture the longer-term

[1] We thank Ran Xu and the Nuance China R&D team for specifying, driving and providing the data collection.

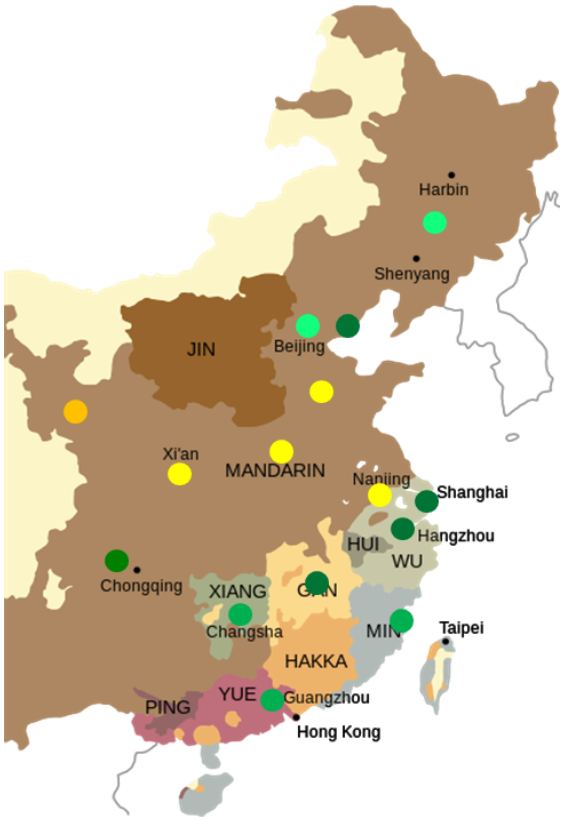


FIGURE 7.1: Locations (indicated by dots) of collection of accented Mandarin speech within Chinese dialect regions.

acoustic context within each speech utterance, which is expected to facilitate accent identification. Furthermore, since it has been shown that the speaker information from i-vector can be used as input feature for DNN acoustic model adaptation [192], we propose to exploit a similar approach for accent identification.

7.3.1.1 i-vector

i-vectors $\mathbf{v}$ are computed from adapted Gaussian Mixture Models (GMMs) with mean super-vector $\mathbf{m}$ and a GMM universal background model (UBM) with mean super-vector $\mathbf{u}$,

$$\mathbf{m} = \mathbf{u} + \mathbf{T}\mathbf{v}, \tag{7.1}$$

where $\mathbf{T}$ is the total "variability matrix" defined in [155]. Similar to [176], the GMMs are trained and adapted using Linear Discriminant Analysis (LDA) based features obtained from sliding windows of acoustic input. In accordance with [181], we use accent independent i-vectors. We performed a preliminary performance evaluation using i-vectors extracted per utterance and a Support Vector Machine for the classification. This only achieved 17.0% accent classification accuracy

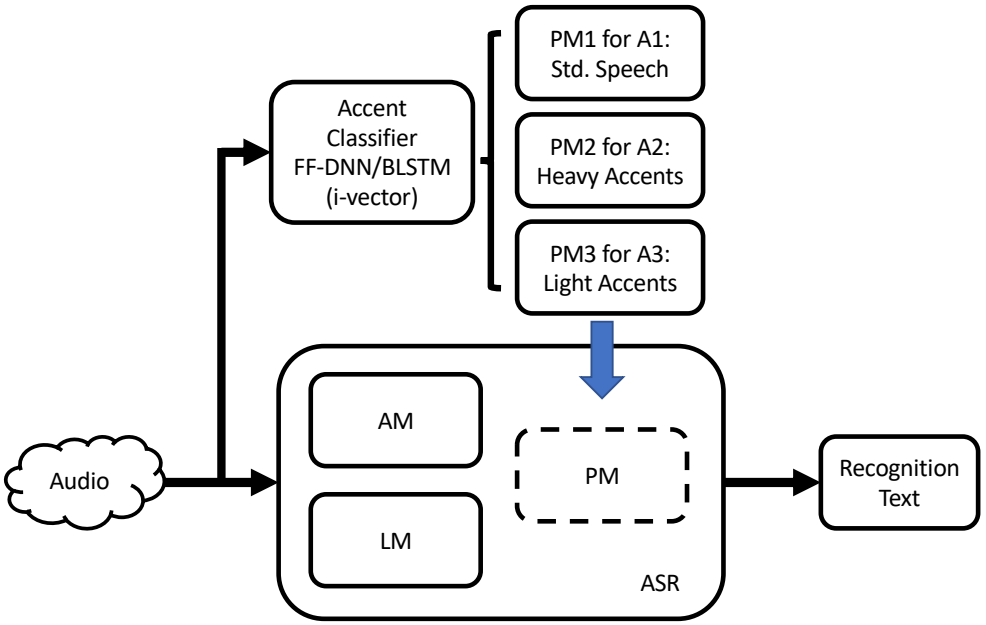


FIGURE 7.2: Flowchart of the ASR system with enhanced Pronunciation Models based on an accent classifier.

(see Table 7.1). Therefore, we decided on the following strategies for increasing performance: (i) computing i-vectors on speaker level, (ii) score addition on speaker level to benefit from cumulative evidence, similar in spirit to [193], and (iii) integration of i-vector into deep learning models. To combine the i-vector with deep learning models, the i-vector is appended to the input features of the DNN or the bLSTM, similar to the method proposed in [192] for speaker adaptation.

7.3.1.2 DNN and bLSTM Accent Classifiers

The acoustic features used to train deep learning accent identification comprise 45 Mel-frequency Cepstral Coefficients (MFCCs) stacked with seven fundamental frequency variation (FFV) features [154] extracted at a frame rate of 10 ms and a window size of 25 ms. The FFV features are added to capture variations of tones in accented Mandarin. The acoustic features used to train deep learning accent identification are the same as for the AMs used for ASR (45 MFCC + 7 FFV features). As a simple deep learning based accent classifier, we explore two feed-forward DNNs, one with and one without i-vector trained on sliding windows of acoustic input, each spanning 63 contiguous frames (645 ms) of speech. The DNNs have two hidden layers of 512 neurons with the rectified linear activation function. The output layer has a softmax activation function and indicates frame-wise posterior probabilities of 15 Mandarin accents and silence, for the center frame

of the input window. The input window size and the DNN topology were chosen empirically based on earlier experiments with speaker identification tasks.

Secondly, we investigate bLSTM classifiers which use single frames of acoustic input. The topology consists of two bLSTM layers of size 512, each comprising 256 LSTM memory cells for the forward and backward directions, followed by the softmax output layer. The topology is designed to have a similar number of parameters as the DNN (2.2M vs. 1.9M). The LSTM cells use the hyperbolic tangent activation function. Training is performed on mini-batches of chunks of 64 contiguous frames. As training algorithms for bLSTM, we explored truncated back-propagation through time (BPTT) and context-sensitive chunk BPTT [194, 195]. For the former, the chunks in each mini-batch are continuations of the chunks in the previous mini-batch, and the forward LSTM states are carried over from one mini-batch to the next. For the latter, the order of chunks is completely random, the LSTM states are reset after each chunk, and the number of contextual frames is set to 16 on each side (i.e., chunks overlap by 50%). Note that bLSTMs trained with truncated BPTT are applied to entire utterances at test time; bLSTMs trained on context-sensitive chunks are also evaluated on complete utterances.

7.3.1.3 Training and Evaluation

The performance of the accent classifiers is evaluated in speaker-independent three-fold cross-validation on the accented Mandarin speech database. The folds are stratified by accent and gender. DNNs and bLSTMs are trained by minimizing the frame-wise cross-entropy loss. The UBM mean vector $\mathbf{u}$ and total variability matrix $\mathbf{T}$ for i-vector estimation (7.1) are calculated on the training set of each fold. In *training*, i-vectors are estimated using all available data per speaker. To improve generalization, i-vectors are randomly dropped out in training. In *testing*, we perform frame- and speaker-level classification. The speaker-level classification is mainly motivated by potential applications of the accent classifier in robust ASR (cf. Chapter 6). It is performed by averaging the posterior probabilities of the 15 accent classes over all frames classified as non-silence. We repeat this procedure utterance by utterance, stopping once the total length of the processed utterances exceeds a given maximum number of frames $T_{\max}$. The test i-vectors are calculated per speaker on the same number of frames. In section 7.3.4 we will explore various

TABLE 7.1: Accent identification accuracy (15 classes) for SVM baseline with speaker i-vectors, and deep learning systems with and without i-vector input. c-bLSTM: bLSTM trained and evaluated on context-sensitive chunks.

Model	i-vector	Accuracy [%]	
		Speaker	Frame
SVM	✓	17.0	–
DNN	–	25.8	13.40
c-bLSTM	–	32.2	16.22
bLSTM	–	32.2	20.74
DNN	✓	**34.1**	21.73
bLSTM	✓	28.5	**26.09**

settings for $T_{\max}$. First, we report on results with $T_{\max} = 6\,000$ frames (one minute of speech).

7.3.2 Performance of Regional Accent Identification

Table 7.1 shows the accuracy of DNN and bLSTM classifiers on the 15-class accent classification task. The frame-level results are given for the 15 accent classes, excluding frames classified as silence. It can be seen that both types of bLSTM models outperform the standard DNN. The bLSTM classifier trained and evaluated on context-sensitive chunks (c-bLSTM) performs considerably worse than standard bLSTM (trained with truncated BPTT and evaluated on entire utterances) on frame level, but not on speaker level. This can be explained by the smoothing effect of preserving the LSTM state between chunks, which, however, does not contribute to the speaker-level accuracy, where multiple frame-level scores are averaged. Furthermore, the performance of c-bLSTM is superior to DNN, although both exploit similar information at the input layer. This in accordance with the findings of [195] obtained on an acoustic modeling task.

The top line in Table 7.1 shows the accuracy obtained in the experiment mentioned in section 7.3.1.1 in which we used a Support Vector Machine to classify accents based on i-vectors. From the Table it is evident that for our task deep learning approaches outperform pure i-vector modeling by a large margin. The addition of i-vector to the DNN input yields a significant improvement and achieves the overall best speaker-level accuracy. We also note that the combination of bLSTM and i-vector leads to additional improvement in frame accuracy. Yet, speaker accuracy is degraded compared to either DNN + i-vector or bLSTM without i-vector; in

fact, the speaker and frame level accuracy are close to each other. We conjecture that the constant i-vector input per frame further adds to the smoothness of the bLSTM output, which has a detrimental effect in the end, since the averaging of the outputs will rarely change the speaker-level result compared to the frame level outputs. This strongly suggests that averaging frame-level scores to obtain speaker-level scores is not the optimal approach. However, an approach based on majority scoring yielded worse speaker-level accuracy.

7.3.3 Error Analysis and Accent Grouping

While the speaker-level accuracy obtained with 15 accent classes (up to 34.1 %) is greatly above chance level (6.7 %), it is still not high enough for accent identification in practical applications. However, a more coarse-grained but more robust classification could be sufficient to identify accented data for accent-specific PM enhancement, and to select appropriate PMs accordingly at test time.

Our hypothesis is that a part of the classification errors can be explained by confusions of the accents of regions which are geographically close to each other, because the Chinese language family consists of a dialect continuum [161]. To confirm this hypothesis, we visualize the class confusions of the bLSTM classifier on speaker level, applying the following method. First, we normalize the confusion matrix $\mathbf{C} = (c_{i,j})$ to sum to one: $\overline{\mathbf{C}} = \mathbf{C} / \sum_{i,j} c_{i,j}$. Then, a symmetric distance matrix $\mathbf{D} = (d_{i,j})$ is obtained as

$$\mathbf{D} = (1 - \overline{\mathbf{C}}) + (1 - \overline{\mathbf{C}})^{\intercal}, \tag{7.2}$$

where $d_{i,j}$ is now a pairwise distance between accent classes i and j. Finally, Kruskal's method for non-metric dimensional scaling (NMDS) [196] is applied to obtain a two-dimensional space where the Euclidean distances between points i and j, representing accent classes, are a monotonic transformation of the distances $d_{i,j}$.

The resulting NMDS configuration is shown in Figure 7.3. Upon closer inspection of Figure 7.3, we can identify three groups of accents that are well separable in the NMDS space: (A1) Beijing, Changchun; (A2) Chengdu, Jinan, Nanjing, Lanzhou, Tangshan, Xi'an, Zhengzhou; (A3) Changsha, Fuzhou, Guangzhou, Hangzhou, Nanchang, Shanghai. Not only are these regions characterized by geographical

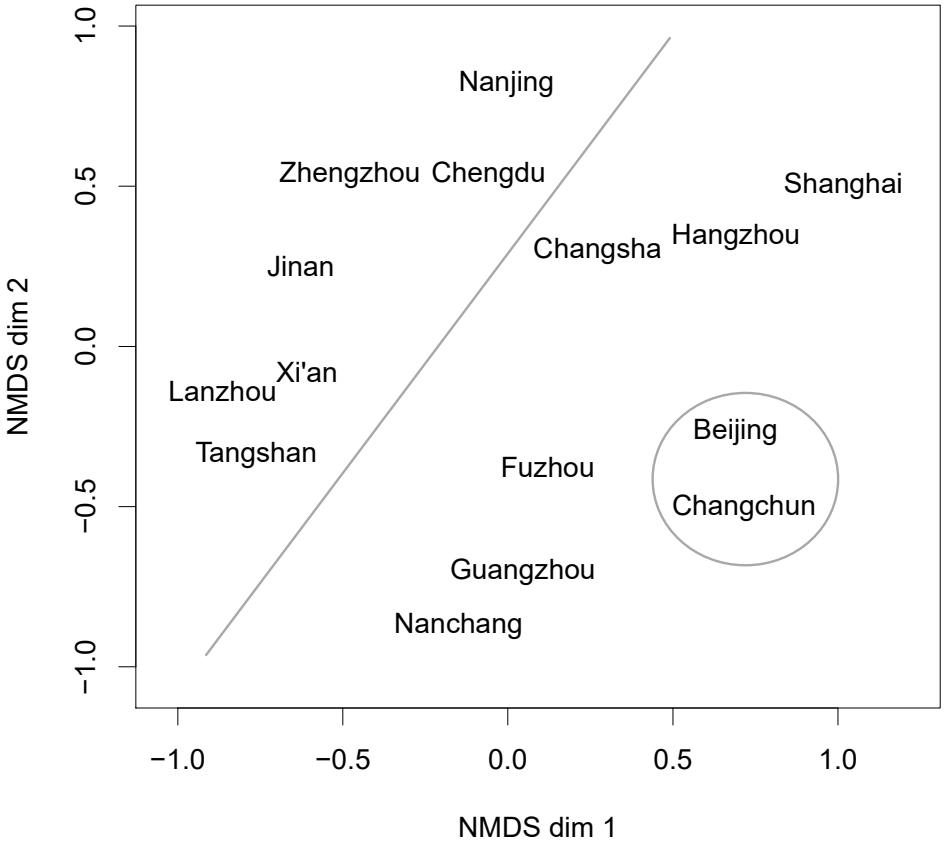
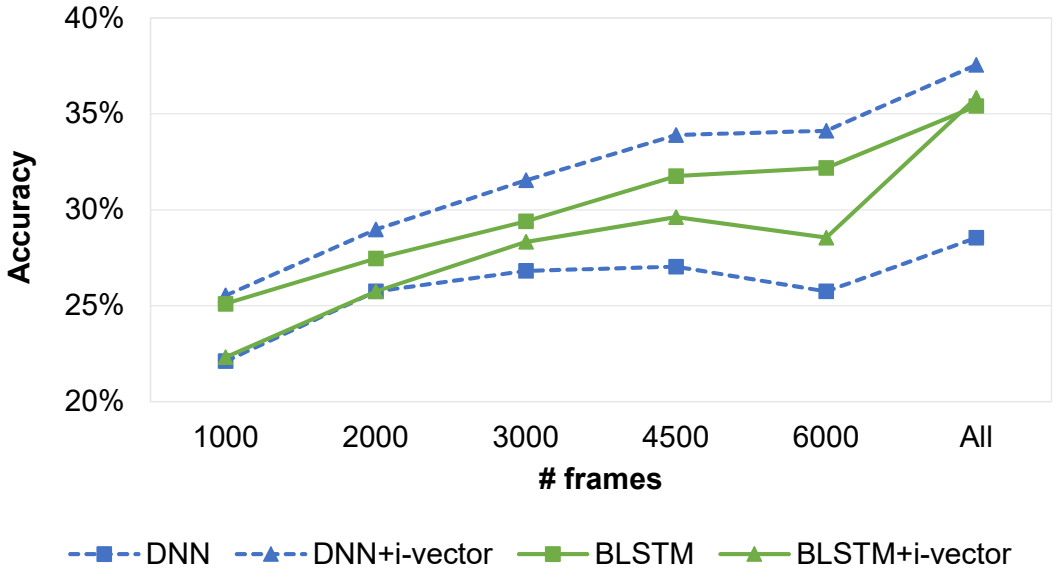


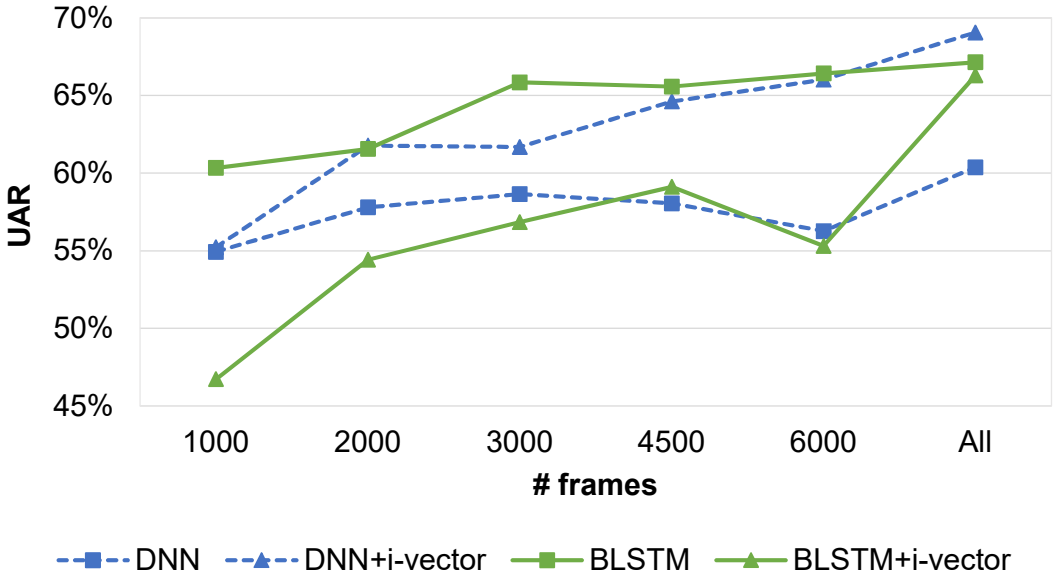
FIGURE 7.3: Non-metric dimensional scaling applied to the distance matrix obtained from the class confusions of the 15-class bLSTM classifier. Gray lines indicate clustering into three accent groups (cf. Section 7.3.3).

proximity (A1: north-east, A2: center-west, A3: south-east), but they can also be interpreted as follows: A1: regions where the local dialect is equal or close to Standard Mandarin; A2: regions where a dialect of Mandarin is the native language; A3: regions where Mandarin is second language.

We can now evaluate the performance of the accent classifiers when mapping the predictions of the 15 accent classes to the groups A1, A2, A3. As performance measure, we opt for unweighted average recall (UAR) [179] since the number of instances is not uniformly distributed across the groups. Based on the predictions of the bLSTM classifier, we obtain 66.4 % UAR. Moreover, the DNN classifier with i-vector input achieves 66.0 % UAR, showing that the accent grouping obtained by analyzing the bLSTM confusions can be generalized.



(A) 15 regional accents



(B) 3 accent groups

FIGURE 7.4: Speaker-level accuracy (top) and unweighted average recall (UAR) (bottom) of accent identification for score averaging as a function of the number of frames per speaker. Note that 'All' corresponds to about 10.9 minutes, i.e., 654,00 frames.

7.3.4 Effect of the Amount of Test Data Per Speaker

For practical applications, it is highly relevant to investigate the performance when only a limited amount of data is available per speaker in testing. Figure 7.4 shows the speaker-level accuracy obtained with varying $T_{\max}$. In case of predicting 15 classes, we see a large impact when varying $T_{\max}$ from 1000 (10 s) to the maximum amount (all frames per speaker, i.e., 10.9 minutes on average). The

best performing classifier (DNN + i-vector) has only 25.5 % accuracy on 10 s of input. For predicting three classes, the bLSTM + i-vector system is most affected by input length, especially when comparing 60 s to all frames. In contrast, the bLSTM without i-vector obtains the best performance (60.3 % UAR) among the classifiers at 10 s of input data, and its performances on 30 s and 60 s of data are very similar. The performance fluctuations when going from 4,500 to 6,000 and then to 650,000 may be related to overfitting in the DNN and the bLSTM+i-vector classifiers. With 654,000 frames those classifiers might be taken into a different regime.[2]

7.4 Accent Robustness of ASR

We now proceed to showing that the accent classifier can be used to select accent specific ASR models for improved robustness. The flowchart of the testing phase is illustrated in Figure 7.1. Our study is based on an ASR system using a hybrid DNN-HMM AM, with optional online speaker adaptation by i-vectors [197]. The AM has more than 20 M weights and is trained on several thousands of hours of collection and field data from the in-car domain. The language model includes an n-gram and a recurrent neural network component and is targeted for the domain of in-car large vocabulary speech recognition. Compared to Chapter 6, both AM and LM are improved by adding in-domain audio and text data without a change of the model size or training strategy.

We use specific pronunciation models for each of the groups A2 and A3, while we use a Standard Mandarin one for the group A1. From knowledge gained in Chapter 6 and from examining ASR outputs, we know that group A2 is mostly characterized by tone confusions. Thus, we designed a specific pronunciation model for the group A2 that includes possible tone confusions (e.g., if speakers from a certain region regularly use the 4^{th} tone instead of the 1^{st} tone, we add 4^{th} tone pronunciations for the 1st tone characters). Similarly, we created a pronunciation model for the group A3 that allows mainly consonant confusions (e.g., *zh* is often pronounced as *z* by these speakers) along with minor tonal ones. The most suitable pronunciation model for each speaker is selected by using the prediction of the 3-class bLSTM accent classifier (without i-vector input, for $T_{\max} = 6\,000$). We

[2] $UEA \approx 67.5\%$ might be the absolute ceiling performance for the three-class task, due to between-speaker variation within the accent groups.

compare this to an oracle experiment where the true accent label is used to switch between pronunciation models.

Table 7.2 shows the CER achieved on the accented Mandarin speech database, subdivided into the 15 locations of collection as well as the three accent groups. As expected, the group A1, whose accent is close to Standard Mandarin, exhibits the lowest CER. Moreover, A2 shows higher CER than A3. An intuitive explanation for this is that speakers from the A3 group consciously switch between Standard Mandarin and their mother tongue, while there is a fuzzy boundary between local dialect and Standard Mandarin in the regions corresponding to A2, which leads to a stronger accent on average.

First, we discuss the results without speaker adaptation. Using the true accent label for model selection, we observe 13 % relative CER reduction (CERR) for the group A2, i.e., the speakers with heavy accents. A similar CERR (11 %) for this group can be obtained when using the predicted accent group, which is notable given the challenge of correct accent identification. However, for the group A3, accent model selection shows no benefit overall. We obtained improvements for part of the speakers in A3, which were, however, canceled by degradation for other speakers. In contrast, we found that the accent model selection helped uniformly for almost all of the heavily accented speakers (A2). On group A1, there is a slight degradation (4.32 to 4.44 % CER) when using the accent prediction instead of the true label, which can be explained by mis-classification of Standard Mandarin speech into one of the accent classes, which causes a less precise pronunciation model to be selected. We could likely avoid some of this degradation by using a confidence threshold for selecting accent specific models, trading in some of the gain on heavily accented speech.

With i-vector speaker adaptation, we obtain relative CERRs of 13.2 %, 15.3 % and 14.6 % for the groups A1, A2 and A3 respectively. This shows that speaker adaptation helps regardless of the accent, as expected. On top of i-vector speaker adaptation, accent model selection notably delivers 8.5 % relative CERR for the group A2. The CER for the A3 and A1 groups is unchanged, which is similar to the case without speaker adaptation.

TABLE 7.2: Character error rate (CER) on accented speech data summarized by accent groups and classes, using ASR systems with and without i-vector speaker adaptation, and additionally with accent model selection using ground truth accent labels (Label) or predictions (Pred) of the 3-class bLSTM classifier.

Spk. adaptation		–	–	–	✓	✓
Model selection		–	Label	Pred	–	Pred
A1	Beijing	4.2	4.2	4.4	**3.7**	3.9
	Changchun	4.4	4.4	4.5	**3.8**	3.8
	Avg.	4.3	4.3	4.4	**3.8**	3.9
A2	Chengdu	5.6	6.2	6.1	**5.1**	5.4
	Jinan	8.8	7.8	7.9	7.6	**7.2**
	Lanzhou	13.7	11.7	12.3	11.4	**10.4**
	Nanjing	9.5	8.3	8.6	7.8	**7.4**
	Tangshan	7.2	6.7	6.7	5.9	**5.7**
	Xi'an	12.5	9.5	10.0	10.5	**8.9**
	Zhengzhou	10.1	8.4	8.6	8.8	**7.8**
	Avg.	9.6	8.4	8.6	8.2	**7.5**
A3	Changsha	6.4	6.3	6.2	5.6	**5.5**
	Fuzhou	5.5	5.5	5.7	**4.8**	5.0
	Guangzhou	5.9	6.0	6.1	**5.1**	5.2
	Hangzhou	6.7	6.5	6.5	**5.6**	5.7
	Nanchang	6.7	6.8	7.0	**5.5**	5.8
	Shanghai	7.0	7.1	7.2	**5.8**	6.1
	Avg.	6.4	6.4	6.4	**5.4**	5.5

7.5 Conclusions

In this study, we have analyzed various approaches based on deep learning and i-vector to identify accented Mandarin speech. The error analysis of accent classification led us to propose a 3-class grouping, which can be used to select accent-specific pronunciation models. We have demonstrated that model switching based on accent prediction can yield CER improvements for a state-of-the art ASR system, even if speaker adaptation is already in place. In future work, we aim to use tone information from ASR to improve accent classification. Moreover, we will explore using the accent classifier to spot accented training data for offline AM adaptation.

Chapter 8

General Discussion and Concluding Remarks

This chapter presents a general discussion of the most important findings from the previous chapters, followed by a discussion of possibilities for further improvement of noise and accent robustness.

Human speech recognition (HSR) is robust against variations of gender, age, articulation rate, accents and noisy backgrounds. HSR is remarkably robust against various types of noise, in a wide range of environments such as streets, airports, restaurants, cars, trains etc. Human recognition performance in noisy environments degrades relatively slowly when the SNR decreases [198]. HSR is also robust against variations in speaking styles. For example, whispering, emotional speech and Lombard effects [199] often do not cause problems for HSR. Concerning accents, people do encounter difficulties with understanding (heavily) accented speech. Nevertheless, human listeners can adapt to accents very quickly [200]. It is usually not a problem for people to be in a conversation with multiple speakers with different accents. The most striking observation is that HSR is effective in virtually all scenarios even when multiple adverse conditions conspire to make understanding difficult.

Although ASR is able to show super-human performance in some tasks [201–203], most ASR systems have not reached the same level of robustness as HSR, especially not in the more challenging conditions mentioned above. In fact, most ASR systems are designed for specific tasks, rather than for a wide range of use. Typically, improving the performance of current ASR systems is achieved by training, tuning

and/or adaptation, using tasks for which the systems were designed, and these improvements seldom generalize to other tasks and conditions. For example, a noise-robust system may perform poorly with clean speech or an accent-robust system may degrade the recognition of accent-free speech. With task-specific performance improvement, versatility is becoming one of the largest gaps between ASR and HSR. This thesis studies two factors that I consider as the major underlying reason for the existing robustness gap between human and automatic speech recognition, namely background noise and accent.

8.1 System Combination for Noise Robustness within a Large SNR Range

It is highly unlikely that humans handle complex and difficult recognition tasks under different conditions with a dedicated strategy optimized for a specific situation. It is well known that human beings compare information from the two ears to localize sources in space and separate sound sources that are arriving from different directions, a process generally known as binaural hearing [204, 205]. It is also well known that hearing-impaired listeners and those listening in adverse acoustic environments (noise, reverberation, multiple speakers) rely heavily on visual input, such as lip movements, as a complementary source of information to the acoustic signal [206]. Also, there is convincing evidence that humans use several different strategies in parallel to solve complex tasks such as speech recognition [207]. As a result, many studies have tried to mimic that human behavior by combining different sources, representations, and modelings of the information in order to improve ASR performance, for example studies with a microphone array [208, 209] and audio-visual ASR systems [210–213] or directly combining different types of ASR systems [51, 214].

The first half of this thesis studies more advanced combination technologies to harness strengths from different systems, targeting – but not constrained to – noise robustness within a large SNR range. The biggest advantage of system combination is that it allows independent and cheaper development of component systems, which can be designed for specific noisy conditions. Our study starts with a small task (i.e., AURORA-2) with the combination of GMM/MLP and SC systems that have been shown to be robust in clean and very noisy conditions, respectively,

and then generalizes to multi-stream deep neural networks on a LVCSR task. I developed a powerful and generic dynamic (i.e., condition dependent and time-variant) weighting scheme based on confidence models at multiple stages in the ASR workflow, including in the feature, probability and lattice domain. Experimental results confirm the advantage of a confidence-based weighting strategy, which yields not only promising recognition accuracies, but also flexibility of extensions to multi-stream combinations.

8.1.1 Larger Differences Lead to Larger Complementarity

8.1.1.1 System Selection

Larger improvements may be expected when individual component systems can be selected that perform significantly better in certain situations than others. Sparse Classification (SC) is chosen because it excels in low SNR conditions compared to other noise robustness techniques [215]. This is thanks to the fact that there are no unseen SNRs in the AURORA-2 corpus from SC's point of view: all additive noisy data is a linear combination of speech and noisy exemplars. Nonetheless, I would still argue that the most compelling reason for the success of the combination involving the SC system is the fact that the representations of the acoustic signals are fundamentally different: a non-parametric representation in the SC system and a parametric in the GMM and MLP baseline systems. (cf. Chapter 2, 3 and 4). This explains why significant improvements are observed not only at very low SNRs, where SC performs very well, but also in those SNR conditions where the SC system is far worse than the baseline systems GMM or MLP. Another example that indicates that larger differences between systems may be beneficial for a successful combination can be found in Chapter 5. A clearly better system `C:FFDNN2` is shown to contribute less when it is combined with a similar system `A:FFDNN` (with exactly the same AM and LM and only differing in terms of operating point) than when it is combined with a weaker stand-alone system `D:LSTM` that is different enough from the AM in both `A:FFDNN` and `C:FFDNN2` regarding the deep learning architecture. Since it is plausible to combine systems that have specific strengths in different situations, an interesting topic for future studies is how complementarity of information can be defined and measured. Besides noise-robust systems that perform very well such as SC, other systems with different specialties might also constitute good candidates in a combination system, such as for example

child-oriented systems, far-talk systems or the accent-specific ones developed in Chapter 6.

8.1.1.2 Stream Transformation

System combination may not work straightforwardly and certain processing of the raw information streams may appear necessary. In Chapter 2, two novel transformation methods are proposed to Gaussianize SC's posterior probability distribution for a better modeling in the Tandem GMM system. Similarly, the SC probability distributions also introduce difficulties in the probability combination studied in Chapter 4. The problem becomes apparent when applying the conventional inverse entropy-based dynamic weighting scheme, and is solved by the newly proposed trustworthiness-based weighting mechanism. The two probability Gaussianization approaches described in Chapter 2 are supposed to transform generic probabilities into Gaussian-like distributions. The trustworthiness-based dynamic weighting method is also meant to be generalizable to a combination of different types of component systems. This is validated in Chapter 5 where the trustworthiness is measured in a different manner, combination occurs in a different stage, the task is more difficult, and the number of component systems is increased from two to five: dynamic weights still provide significant gains over static ones. Likely, the proposed methods are applicable directly to many other situations. If they are not directly applicable, I would still suggest to investigate a proper transformation or combination method to integrate newly proposed component streams that are believed to contain complementary knowledge, rather than move on to find easier-to-be-combined sources or systems which may have less complementarity and will likely end up with a mediocre combination.

8.1.2 How to combine

8.1.2.1 Dynamic Weights

Another issue that emerged from the attempts to cover a very wide SNR range by combining multiple systems is the need for dynamic weighting. This is because the accuracy of the response of a given system to an arbitrary unseen utterance is difficult to predict on the basis of a global estimate of the conditions under

which the utterance was recorded. It appears that unseen utterances can have idiosyncratic characteristics that make it easier for one system than for another, independent of global measures such as SNR. It would be very helpful, therefore, if it were possible to derive measures from each utterance to predict the accuracy of several systems for each utterance. These accuracies can subsequently be used when combining the systems. This corresponds to the finding that humans may use local rather than global features in their processing of perceptual input [216].

System performance is predicted by confidence models. Static weights used in Chapter 3 can be considered as a confidence model with a constant output. In Chapter 4, the confidence model is represented by a look-up table that describes the relationship between the entropy of a probability vector and the trustworthiness of the decoding of an unknown utterance. Finally in Chapter 5, a confidence model was developed in the form of a logistic regression model that was trained for each component system separately. Although the information to be combined is conceptually different across these three chapters mentioned above (i.e., the integration of SC via virtual evidence in a DBN, the fusion of probability vectors via a switch between summation and multiplication and finally an off-line lattice combination), in all cases it appeared that dynamically adapting the weights attached to the streams outperformed static weights, regardless of how those static weights were estimated.

Further improvement of the dynamic weights may come from weight smoothing. Given the assumption that the test conditions, including the speaker, channel and background noise, will be relatively stable across neighboring frames or utterances, the combination weights are likely to benefit from smoothing over time in real applications. Especially for the dynamic weights within an utterance the confidence estimates may differ dramatically between silence and speech frames, which results in spurious fluctuations over time. An interesting topic for future study is whether further improvements can be obtained from smoothing the dynamic weights, for example by applying a regularization factor or averaging weights over a sliding window.

Another interesting topic for further research is joint training of the confidence models for all component systems. As mentioned in Chapter 4, it is important to have separate estimates of confidence to cope with potentially different properties among component systems. The key is separate confidence measures. Although the confidences used in this thesis are bounded between 0 and 1 and are normalized

to sum up to 1, it could still be helpful to add calibrations of the weights to have a better balance. This can be achieved either by adding an optimized bias or via a joint estimate of the weight set. A joint estimate of the weight set may be particularly interesting since it directly measures the relative importance, which essentially matters in system combinations. Note that joint estimation of confidence scores is different from using one single confidence criterion. The latter applies the same rule (for example using inverse entropy) to all systems, whereas the former estimates different systems jointly in one shot. In fact, it is similar to the auto-encoder proposed in [217].

Combination weights estimated in this thesis are based on system confidence scores: a probability of recognition correctness that rarely has hard zeros, so that generally all component systems play a role in the final combination. However, allowing all systems to play a role –though sometimes a very small one– may not always be the best option. It is observed that the collapse of one component system will lead to a vulnerable combined system in the dynamic combination approach study between SUM and PRODUCT in Chapter 4. It would be interesting to have a confidence model that can act as a circuit breaker: that can detect and disable a low-performing component systems to protect the combination from single-system failures. Possibly, one can start with some thresholds as a post-processing of the confidence scores, and mute systems scoring below the threshold.

8.1.2.2 Dynamic Combination Methods

After determining the weight set for all component streams to be combined, the combination method must also be optimized. Different combination methods have different strengths. Taking the summation and multiplication methods as an example, summation is less aggressive in changing the recognition hypothesis of the combination than multiplication, which makes summation more robust against component system failures. In Chapter 4, a dynamic arbitration between these two combination approaches is found to be beneficial. The switch is based on the dot product of two component streams. In Chapter 5, two new combination methods are compared in the lattice domain, CNC and MBR. Though MBR yields much better performance than CNC, it is well known that CNC is more computationally efficient than MBR. As a result, it is interesting to investigate in which situations CNC might be sufficient for recognition, so that the more expensive

MBR solution can be avoided. Future studies should investigate automatic decisions about combination methods, such as a neural network solution that builds all component systems jointly and integrates the learning methods for selecting the most promising combination, rather than implementing a hard switch decision after each component stream has been processed.

8.1.3 Combination Technologies beyond Noise-robust ASR

The proposed confidence-based dynamic weighting scheme can be useful in other applications than noise-robust ASR. For example, the same idea could be applied in -for example- audio-visual and microphone-array systems. It would require dynamically estimating the local confidence per component system so that the confidence scores can be used as weights in the combination. The dynamic weighting techniques that are developed as part of the research in this thesis for probability or lattice combination may very well generalize to other intermediate stages of information processing or even to completely different types of systems. End-to-end systems are trending nowadays, where 'Attention' has become the core element since the first Google paper published [218]. A weighting scheme can be applied at multiple places in those kinds of systems. For example, it can be applied to lattices generated by multiple end-to-end systems in the same manner as in Chapter 5.

The weighting scheme could also be applied to attention scores from different encoders or different sub-spaces of one single encode,r such as the multi-head attention system. Actually, various examples exist where the weighting scheme developed in this thesis is evolving into an attention mechanism. For instance, a so-called hierarchical attention network was introduced in [217] to allocate dynamic weights for multi-microphone AMs, a similar attention-based alignment network called Gated Bidirectional Alignment Network (GBAN) is introduced for multi-modal emotion recognition in [219], and multi-stream convolution neural networks (CNNs) are fused via a self-attentive simple recurrent unit (SRU) in [220]. Additionally, another hot research topic is the joint use of connectionist temporal classification (CTC) and attention loss functions, which provide large improvements to the latest end-to-end systems [221–223]. Such systems may also benefit from an adaptive balance between different losses in training. In short, the fusion technology proposed in this thesis can play a role when different voices fuse within a single or from several ASR systems.

8.2 Robustness against Multiple Accents

Accent robustness is a serious challenge that has been recognized in the literature and by companies that market real-world applications of ASR. The difficulty is caused not only by deviations in pronunciation, but also by different vocabularies and even grammars that are used by speakers with different language and accent backgrounds. In the second part of the thesis several approaches are studied to address the difficulties caused by accented spoken words in a 'standard' grammar. Most previous studies focused on improving recognition performance for a specific accent, and ignored any performance changes on the standard accent or on other non-standard accents. This lack of versatility is probably the most striking difference between ASR and HSR and explains to a large extent why until now ASR systems have not been very successful in comprehending multiple accents in a single conversation. This is in strong contrast to humans who can adapt to an accent quickly, even if it is an unknown one. In this thesis, a real-time ASR system is investigated that can tackle multiple accents including the standard one at the same time.

8.2.1 Speaker Adaptation for Accent Robustness

The i-vector, which is commonly used to capture characteristics at the speaker level, implicitly also covers accent variation. Experimental results in Chapter 7 show that speaker adaptation achieved via i-vector as an auxiliary input yields significant gains regardless of the accent. The results suggest that it is worthwhile to conduct future studies using on-line or off-line adaptation methods, such as Constrained Maximum Likelihood Linear Regression (CMLLR) [224] or the i-vector's successor "x-vector" [225]. Another promising idea is to exploit more accent-specific characteristics directly. For example, accent embedding vectors can be extracted from an intermediate hidden layer of a DNN that is trained on accent targets. This idea, implemented in a recent study [226], shows a better clustering of the accents than i-vectors in a 2D t-SNE projection [227] and impressive gains on recognition of four English accents accordingly.

8.2.2 Lexicon-based Accent Enhancement

8.2.2.1 DNN-based Approach for Accent Enhancement

In Chapter 6, two ways are proposed to discover suitable phonetic confusion rules for enhancing the lexicon. Although the performance levels are promising, both methods constitute a rule-based approach that is less flexible and more likely to result in a sub-optimal compromise. The best system using syllable-based context-dependent confusions is a good example of a system that measures both the gains and losses of individual syllable confusions and subsequently only selects the ones introducing big gains and small losses. The ones that result in big gains and big losses will likely be eliminated due to the potential risk of degradations. Instead of statistical counting, an interesting research direction is to build a confusion classifier in the DNN framework, with valid acoustic features as its input and the word-level accent or syllable labels as its target. In such a way, it would not be necessary to predict the accent any longer. Instead, the set of phonetic confusions could be used directly. This would allow the accent enhancement to be estimated dynamically based on current input data, without ruling out confusions that can yield mixed results.

8.2.2.2 Should accent be handled by AM or PM?

Both the AM and PM approach are tried out in Chapter 6 to increase accent robustness. Although both show promising results, I would like to elaborate a bit on the difference. More specifically, on the question of whether accent should be tackled by AM or PM. By definition, an AM is a model that is supposed to predict how a word is composed of a sequence of phones or other word or sub-word units. This is independent from accent, even from language in some cases. This is the reason why it is feasible to develop language-independent AMs such as [228–230]. Essentially, the accent-robust AM proposed in Chapter 6 is a weaker model of standard speech, with blurred boundaries between different acoustically similar units. This approach runs counter to the goal of good acoustic modeling, which is accurate modeling of pre-defined acoustic units. Besides the expected degradation for standard speech, another common difficult situation that AMs cannot solve is that the pronunciation of some tokens according to accent 'A' may be still legitimate, but refer to something different in accent 'B'. For example, the word

"Holland(荷兰)" is pronounced as "He2 Lan2" in standard Mandarin, while it is commonly (mis)pronounced in a wide southern area of China as "He2 Nan2", which is identical to the pronunciation of another word "HeNan(河南)", a province in the middle of China. Acoustically, there is no difference between the pronunciations of these two different tokens, one of standard and the other of accented Mandarin. Therefore, I would argue that it is better to let a more versatile PM handle such phonetic changes rather than to introduce a weak AM.

8.2.3 Accent Classification for Multiple Accent Support

In Chapter 6, a universal lexicon modification is proposed that aims to provide a good enhancement across 15 Mandarin accents. The goal is achieved by introducing a context-dependent syllable level, measuring the gains and losses for each syllable confusion and finally selecting those syllable candidates which mostly "add" rather than "subtract" in the final confusions set for a lexicon modification. This approach is superior for most accents, even better than some accent-specific approaches. Still, it is clearly sub-optimal for individual accent performance, because the selection of syllable confusions does not only depend on whether they will lead to improvements for a certain accent, but the degradations they cause for other accents must also be considered at the same time. Obviously, the more accents involved, the more compromises the approach will need to make.

Chapter 7 provides an accent classifier hat serves as an accent pre-selector that allows designing more versatile accent-specific lexicons. Despite of the gains shown in Table 7.2, building such an accent classifier is not a trivial task. In this section, several topics are discussed regarding ways in which an accent classifier can further contribute to multiple-accent ASR and beyond.

8.2.3.1 How to Improve the Accent Classifier

Due to the issue presented in Section 8.2.2.2, it is clear that acoustic inputs are not enough for a good accent classifier. A larger window of the semantic information at the sentence level is necessary. Taking the "Holland"" and "HeNan" example again, if 'Europe' or 'stroopwafels' is mentioned somewhere in the same sentence or context – according to the ASR engine – then "Holland" is more likely to be the correct recognition hypothesis than "HeNan", which will win in the context

of “China” or “chicken feet”. Consequently, the accent prediction will be standard Mandarin rather than southern Chinese accents. An interesting topic for future studies is a smart design of a joint ASR, natural language understanding and accent classifier that involves either a second recognition pass or the semantic attention mechanism.

8.2.3.2 Difficulty: How to Define an Accented Sentence

In both Chapter 6 and 7, categories of accents are used in reporting ASR results. Next to the difference concerning the single light/standard accent groups in Chapter 6 being split as separate standard and light accent groups in Chapter 7, another particular difference is the swapping of the ChangSha and ChengDu accents. This observation leads to the question how humans perceive accents. When manually categorizing the accents for the work in Chapter 6, the language developer observed that utterances require only very few non-standard words to be regarded as a heavily accented, as long as the deviation of the pronunciation is obvious enough. At the same time, this manual accent categorization seems more in line with the ASR performance with heavy accent enhancements than the one in Chapter 7. It can be visualized in the corresponding heat-map plots in Fig 6.4 and 6.5.

The classification methods proposed in Chapter 7 are based on majority votes at utterance or speaker level. Since it is very rare to see two different accents (other than the standard one) in one single utterance from the same speaker, and while only few words in an utterance may be heavily accented, accent spotting might become a more suitable method for future investigations. If accent labels are assigned at the word level instead of utterance level, it seems like a good idea to generate them based on the ASR results for training. Table 6.6 shows the large changes of CER that can be achieved by different accent enhancements: big losses with standard data and big wins with heavy accented data. A standard ASR system can be trained to be accurate for standard speech but vulnerable to mis-recognitions with accented speech. Since the ground-truth transcription is given for training data, the alignment of its standard recognition results with the one from the accent-enhanced system can tell which words are likely to have a certain accent and which are not: ‘accented’ if the latter recognition is correct but the former is wrong; otherwise ‘standard’ to be conservative. That would be a way to generate the word-level accent labels automatically.

8.2.3.3 Accent Classifier Contributions beyond ASR

Dialogue systems that use ASR are increasingly being deployed in a variety of business and enterprise applications. Moreover, there has been a shift from one-way command-based dialog systems to two-way conversational systems, that are usually composed of an ASR system as the human-to-machine interface, a Natural Language Understanding and Generation unit as the semantic brain and a Text-To-Speech component as the machine-to-human interface. Besides the accent-robust ASR that is studied in this thesis, the other three components can benefit from a well-built accent classifier as well. This is mainly because information about a distinguished accent can be used as information about the users' background that can be utilized for personalization. This idea is described in more detail in a patent application that I filed in 2019, entitled "System and Methods for Accent Classification" [231].

Bibliography

[1] J. F. Gemmeke, T. Virtanen, and A. Hurmalainen. Exemplar-based sparse representations for noise robust automatic speech recognition. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 19(7): 2067–2080, 2011.

[2] E. Vincent, J. Barker, S. Watanabe, J. Le Roux, F. Nesta, and M. Matassoni. The second 'CHiME' speech separation and recognition challenge: An overview of challenge systems and outcomes. In *Proc. of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 162–167, 2013.

[3] Jon Barker, Ricard Marxer, Emmanuel Vincent, and Shinji Watanabe. The third 'CHiME' speech separation and recognition challenge: Dataset, task and baselines. In *Proc. of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 504–511, 2015.

[4] Mirco Ravanelli, Philemon Brakel, Maurizio Omologo, and Yoshua Bengio. A network of deep neural networks for distant speech recognition. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4880–4884, 2017.

[5] Damjan Vlaj and Zdravko Kačič. The influence of Lombard effect on speech recognition. *Speech Technologies*, pages 1998–2001, 2011.

[6] Ricard Marxer, Jon Barker, Najwa Alghamdi, and Steve Maddock. The impact of the Lombard effect on audio and visual speech recognition systems. *Speech Communication*, 100:58–68, 2018.

[7] Daniel Michelsanti, Zheng-Hua Tan, Sigurdur Sigurdsson, and Jesper Jensen. Deep-learning-based audio-visual speech enhancement in presence of Lombard effect. *Speech Communication*, 115:38–50, 2019.

[8] Richard Wright. Intra-speaker variation and units in human speech perception and ASR. In *Speech Recognition and Intrinsic Variation Workshop*, 2006.

[9] Thibault Viglino, Petr Motlicek, and Milos Cernak. End-to-End Accented Speech Recognition. In *Proc. of the International Speech Communication Association (INTERSPEECH)*, pages 2140–2144, 2019.

[10] Genta Indra Winata, Samuel Cahyawijaya, Zihan Liu, Zhaojiang Lin, Andrea Madotto, Peng Xu, and Pascale Fung. Learning fast adaptation on cross-accented speech recognition. *arXiv preprint arXiv:2003.01901*, 2020.

[11] H. G. Hirsch and D. Pearce. The Aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions. In *Proc. International Conference on Spoken Language Processing (ICSLP)*, pages 29–32, Beijing, China, 2000.

[12] Zixing Zhang, Jürgen Geiger, Jouni Pohjalainen, Amr El-Desoky Mousa, Wenyu Jin, and Björn Schuller. Deep learning for environmentally robust speech recognition: An overview of recent developments. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 9(5):1–28, 2018.

[13] M.J.F. Gales. Maximum likelihood linear transformations for HMM-based speech recognition. *Computer Speech and Language*, 12(2):75 – 98, 1998.

[14] Daniel Povey and George Saon. Feature and model space speaker adaptation with full covariance Gaussians. In *Ninth International Conference on Spoken Language Processing*, 2006.

[15] Andrea Schnall and Martin Heckmann. Feature-space SVM adaptation for speaker adapted word prominence detection. *Computer Speech & Language*, 53:198–216, 2019.

[16] András Zolnay, Ralf Schlüter, and Hermann Ney. Robust speech recognition using a voiced-unvoiced feature. In *Seventh International Conference on Spoken Language Processing*, 2002.

[17] Ralf Schluter, Ilja Bezrukov, Hermann Wagner, and Hermann Ney. Gammatone features and feature combination for large vocabulary speech recognition. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 4, pages IV–649, 2007.

[18] Christoph Lüscher, Eugen Beck, Kazuki Irie, Markus Kitza, Wilfried Michel, Albert Zeyer, Ralf Schlüter, and Hermann Ney. RWTH ASR systems for librispeech: Hybrid vs attention–w/o data augmentation. *arXiv preprint arXiv:1905.03072*, 2019.

[19] Daria Vazhenina and Konstantin Markov. End-to-end noisy speech recognition using Fourier and Hilbert spectrum features. *Electronics*, 9(7):1157, 2020.

[20] H. Misra, H. Bourlard, and V. Tyagi. New entropy based combination rules in HMM/ANN multi-stream ASR. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 2, pages II–741, 2003.

[21] Sri Harish Mallidi, Tetsuji Ogawa, Karel Veselỳ, Phani S Nidadavolu, and Hynek Hermansky. Autoencoder based multi-stream combination for noise robust speech recognition. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[22] Xiang Li, Rita Singh, and Richard M Stern. Lattice combination for improved speech recognition. In *Proc. International Conference on Spoken Language Processing (ICSLP)*, 2002.

[23] Haihua Xu, Daniel Povey, Lidia Mangu, and Jie Zhu. Minimum Bayes risk decoding and system combination based on a recursion for edit distance. *Computer Speech & Language*, 25(4):802–828, 2011.

[24] Mahesh Kumar Nandwana, Julien van Hout, Colleen Richey, Mitchell McLaren, Maria Auxiliadora Barrios, and Aaron Lawson. The VOiCES from a distance challenge 2019. In *Proc. of the International Speech Communication Association (INTERSPEECH)*, pages 2438–2442, 2019.

[25] J. G. Fiscus. A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER). In *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pages 347–354, 1997.

[26] Takuya Yoshioka, Dimitrios Dimitriadis, Andreas Stolcke, William Hinthorn, Zhuo Chen, Michael Zeng, and Xuedong Huang. Meeting transcription using asynchronous distant microphones. In *Proc. of the International Speech Communication Association (INTERSPEECH)*, pages 2968–2972, 2019.

[27] Zoltán Tüske, Kartik Audhkhasi, and George Saon. Advancing sequence-to-sequence based speech recognition. In *Proc. of the International Speech Communication Association (INTERSPEECH)*, pages 3780–3784, 2019.

[28] Mirjam Wester. Pronunciation modeling for ASR–knowledge-based and data-derived methods. *Computer Speech & Language*, 17(1):69–85, 2003.

[29] Jason J Humphries, Philip C Woodland, and D Pearce. Using accent-specific pronunciation modelling for robust speech recognition. In *Proc. International Conference on Spoken Language Processing (ICSLP)*, volume 4, pages 2324–2327, 1996.

[30] Dimitra Vergyri, Lori Lamel, and Jean-Luc Gauvain. Automatic speech recognition of multiple accented English data. In *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[31] Yi-Chen Chen, Zhaojun Yang, Ching-Feng Yeh, Mahaveer Jain, and Michael L Seltzer. Aipnet: Generative adversarial pre-training of accent-invariant networks for end-to-end speech recognition. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6979–6983, 2020.

[32] Santiago Omar Caballero Morales and Stephen J Cox. Modelling errors in automatic speech recognition for dysarthric speakers. *EURASIP Journal on Advances in Signal Processing*, 2009(1):308340, 2009.

[33] Woo Kyeong Seong, Ji Hun Park, and Hong Kook Kim. Dysarthric speech recognition error correction using weighted finite state transducers based on context–dependent pronunciation variation. In *International Conference on Computers for Handicapped Persons*, pages 475–482. Springer, 2012.

[34] Emre Yılmaz, Vikramjit Mitra, Ganesh Sivaraman, and Horacio Franco. Articulatory and bottleneck features for speaker-independent ASR of dysarthric speech. *Computer Speech & Language*, 58:319–334, 2019.

[35] Helmer Strik and Catia Cucchiarini. Modeling pronunciation variation for ASR: A survey of the literature. *Speech Communication*, 29(2):225 – 246, 1999.

[36] Hynek Hermansky, Daniel PW Ellis, and Sangita Sharma. Tandem connectionist feature extraction for conventional HMM systems. In *Proc. of*

IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), volume 3, pages 1635–1638, 2000.

[37] Dong Yu and Michael L Seltzer. Improved bottleneck features using pretrained deep neural networks. In *Proc. of the International Speech Communication Association (INTERSPEECH)*, 2011.

[38] Jort F Gemmeke, Tuomas Virtanen, and Antti Hurmalainen. Exemplar-based sparse representations for noise robust automatic speech recognition. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 19(7):2067–2080, 2011.

[39] Zhizheng Wu, Tuomas Virtanen, Eng Siong Chng, and Haizhou Li. Exemplar-based sparse representation with residual compensation for voice conversion. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 22(10):1506–1521, 2014.

[40] S. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 28(4): 357–366, 1980.

[41] Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988.

[42] Jeff Bilmes and Geoff Zweig. The graphical models toolkit: An open source software system for speech and time-series processing. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2002.

[43] Simon King, Joe Frankel, Karen Livescu, Erik McDermott, Korin Richmond, and Mirjam Wester. Speech production knowledge in automatic speech recognition. *The Journal of the Acoustical Society of America*, 121(2):723–742, 2007.

[44] David L Thomson and Rathinavelu Chengalvarayan. Use of periodicity and jitter as speech recognition features. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 1, pages 21–24, 1998.

[45] András Zolnay, Ralf Schluter, and Hermann Ney. Acoustic feature combination for robust speech recognition. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 1, pages I–457, 2005.

[46] Van-Thuan Tran and Wei-Ho Tsai. Acoustic-based emergency vehicle detection using convolutional neural networks. *IEEE Access*, 8:75702–75713, 2020.

[47] Sangita Sharma. *Multi-stream approach to robust speech recognition.* PhD thesis, Oregon Graduate Institute of Science and Technology, 1999.

[48] Sridhar Krishna Nemala, Kailash Patil, and Mounya Elhilali. A multistream feature framework based on bandpass modulation filtering for robust speech recognition. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 21(2):416–426, 2012.

[49] Hynek Hermansky. Coding and decoding of messages in human speech communication: Implications for machine recognition of speech. *Speech Communication*, 106:112–117, 2019.

[50] Hemant Misra, Hervé Bourlard, and Vivek Tyagi. New entropy based combination rules in HMM/ANN multi-stream ASR. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 741–744, 2003.

[51] Christian Plahl, Michael Kozielski, Ralf Schlüter, and Hermann Ney. Feature combination and stacking of recurrent and non-recurrent neural networks for LVCSR. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6714–6718, 2013.

[52] Gunnar Evermann and P.C. Woodland. Posterior probability decoding, confidence estimation and system combination. In *Proc. of Speech Transcription Workshop*, volume 27, pages 78–81. Baltimore, 2000.

[53] Mingkuan Liu and Bo Xu. Accent-specific Mandarin adaptation based on pronunciation modeling technology. In *Proceedings of Interspeech*, pages 330–333, 2000.

[54] Steven B Davis and Paul Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences.

Acoustics, Speech and Signal Processing, IEEE Transactions on, 28(4):357–366, 1980.

[55] Hervé Bourlard, Hynek Hermansky, and Nelson Morgan. Towards increasing speech recognition error rates. *Speech Communication*, 18:205–231, 1996.

[56] Hervé Bourlard, Hynek Hermansky, and Nelson Morgan. Towards increasing speech recognition error rates. *Speech Communication*, 18:205–231, 1996.

[57] Hynek Hermansky, Brian A Hanson, and Hisashi Wakita. Perceptually based linear predictive analysis of speech. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 10, pages 509–512, 1985.

[58] Hynek Hermansky. Perceptual linear predictive (PLP) analysis of speech. *the Journal of the Acoustical Society of America*, 87(4):1738–1752, 1990.

[59] Hynek Hermansky and Nelson Morgan. RASTA processing of speech. *Speech and Audio Processing, IEEE Transactions on*, 2(4):578–589, 1994.

[60] Sara Ahmadi, Seyed Mohammad Ahadi, Bert Cranen, and Lou Boves. Sparse coding of the modulation spectrum for noise-robust automatic speech recognition. *EURASIP Journal on Audio, Speech, and Music Processing*, 36:1 – 20, 2014.

[61] Sara Ahmadi, Bert Cranen, Lou Boves, Louis ten Bosch, and Antal van den Bosch. Human-inspired modulation frequency features for noise-robust ASR. *Speech Communication*, 84:66 – 82, 2016.

[62] Jibin Wu, Emre Yılmaz, Malu Zhang, Haizhou Li, and Kay Chen Tan. Deep spiking neural networks for large vocabulary automatic speech recognition. *Frontiers in Neuroscience*, 14:199, 2020.

[63] Herve A Bourlard and Nelson Morgan. *Connectionist speech recognition: a hybrid approach*, volume 247. Springer Science & Business Media, 2012.

[64] Hynek Hermansky, Daniel P. W. Ellis, and Sangita Sharma. Tandem connectionist feature extraction for conventional HMM systems. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1635–1638, 2000.

[65] Daniel Ellis, Rita Singh, and Sunil Sivadas. Tandem acoustic modeling in large-vocabulary recognition. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 517–520, 2001.

[66] Frank Seide, Gang Li, and Dong Yu. Conversational speech transcription using context-dependent deep neural networks. In *Proc. of the International Speech Communication Association (INTERSPEECH)*, pages 437–440, 2011.

[67] Tara N. Sainath, Brian Kingsbury, Bhuvana Ramabhadran, Petr Fousek, Petr Novák, and Abdel rahman Mohamed. Making deep belief networks effective for large vocabulary continuous speech recognition. In *Proc. of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 30–35, 2011.

[68] Tara N. Sainath, Brian Kingsbury, Abdel rahman Mohamed, George E. Dahl, George Saon, Hagen Soltau, Tomás Beran, Aleksandr Y. Aravkin, and Bhuvana Ramabhadran. Improvements to deep convolutional neural networks for LVCSR. In *Proc. of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 315–320, 2013.

[69] Jort F. Gemmeke, Tuomas Virtanen, and Antti Hurmalainen. Exemplar-based sparse representations for noise robust automatic speech recognition. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 19(7):2067–2080, 2011.

[70] Katariina Mahkonen, Antti Hurmalainen, Tuomas Virtanen, and Jort F. Gemmeke. Mapping sparse representation to state likelihoods in noise-robust automatic speech recognition. In *Proc. of the International Speech Communication Association (INTERSPEECH)*, pages 465–468, 2011.

[71] Noboru Kanedera, Takayuki Arai, Hynek Hermansky, and Misha Pavel. On the relative importance of various components of the modulation spectrum for automatic speech recognition. *Speech Communication*, 28(1):43 – 55, 1999.

[72] J. Gemmeke, Antti Hurmalainen, Tuomas Virtanen, and Y. Sun. Toward a practical implementation of exemplar-based noise robust ASR. In *Proc. of European Signal Processing Conference (EUSIPCO)*, Barcelona, Spain, 2011.

[73] J. Gemmeke, L. ten Bosch, L.Boves, and B. Cranen. Early fusion of sparse classification and GMM for noise robust ASR. In *Proc. of European Signal Processing Conference (EUSIPCO)*, Glasgow, Scotland, 2009.

[74] Jort F. Gemmeke, Tuomas Virtanen, and Antti Hurmalainen. Exemplar-based speech enhancement and its application to noise-robust automatic speech recognition. In *International Workshop on Machine Listening in Multisource Environments*, pages 1–6, 2011.

[75] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

[76] Yang Sun, Bert Cranen, Jort F. Gemmeke, Lou Boves, Louis ten Bosch, and Mathew M. Doss. Using sparse classification outputs as feature observations for noise-robust ASR. In *Proc. of the International Speech Communication Association (INTERSPEECH)*, 2012.

[77] J. de Veth, B. Cranen, and L. Boves. Acoustic backing-off as an implementation of missing feature theory. *Speech Communication*, 34:247 –265, 2001.

[78] Florian Erich Hilger. *Quantile based histogram equalization for noise robust speech recognition.* PhD thesis, RWTH Aachen, 2004.

[79] J.F. Gemmeke and T. Virtanen. Noise robust exemplar-based connected digit recognition. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Dallas, TX, 2010.

[80] Solomon Kullback. *Information Theory and Statistics.* Dover Publications Inc., 1968.

[81] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *Proc. of Neural Information Processing Systems*, pages 556–562, 2000.

[82] Daniel P. W. Ellis and Manuel J. Reyes Gomez. Investigations into tandem acoustic modeling for the Aurora task. In *Proc. of the International Speech Communication Association (INTERSPEECH)*, pages 189–192, 2001.

[83] Yifan Gong. Speech recognition in noisy environments: A survey. *Speech Communication*, 16(3):261–291, 1995.

[84] Yang Sun, Jort F. Gemmeke, Bert Cranen, Louis ten Bosch, and Lou Boves. Early fusion of sparse classification and GMM for noise robust ASR. In *Proc. of European Signal Processing Conference (EUSIPCO)*, pages 1495–1499, Barcelona, Spain, 2011.

[85] Yang Sun, Jort F. Gemmeke, Bert Cranen, Louis ten Bosch, and Lou Boves. Fusion of parametric and non-parametric approaches to noise-robust ASR. *Speech Communication*, pages 49–62, 2014.

[86] Hugo Van Hamme. PROSPECT features and their application to missing data techniques for robust speech recognition. In *Proc. of the International Speech Communication Association (INTERSPEECH)*, 2004.

[87] ETSI. ETSI standard doc.: Speech processing, transmission and quality aspects (STQ); distributed speech recognition; advanced front-end feature extraction algorithm;, 2007.

[88] Yang Sun, Mathew M. Doss, Jort F. Gemmeke, Bert Cranen, Louis ten Bosch, and Lou Boves. Combination of sparse classification and multilayer perceptron for noise-robust ASR. In *Proc. of the International Speech Communication Association (INTERSPEECH)*, pages 310–313, 2012.

[89] B. Raj and R.M. Stern. Missing-feature approaches in speech recognition. *IEEE Signal Processing Magazine*, 22(5):101–116, 2005.

[90] M.L. Seltzer, B. Raj, and R.M. Stern. A Bayesian classifier for spectrographic mask estimation for missing feature speech recognition. *Speech Communication*, 43(4):379–393, 2004.

[91] M. Cooke, P. Green, L. Josifovski, and A. Vizinho. Robust automatic speech recognition with missing and unreliable acoustic data. *Speech Communication*, 34(3):267–285, 2001.

[92] R.C. van Dalen and M.J.F. Gales. Extended VTS for noise-robust speech recognition. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 19(4):733–743, 2011.

[93] Ramya Rasipuram and Mathew Magimai Doss. Integrating articulatory features using Kullback-Leibler divergence based acoustic model for phoneme recognition. In *Proc. of IEEE International Conference on Acoustics, Speech*

and Signal Processing (ICASSP), pages 5192 – 5195, Prague, Czech Republic, 2011.

[94] Guillermo Aradilla. *Acoustic models for posterior features in speech recognition*. PhD thesis, Ecole Polytechnique Fédérale de Lausanne, Switzerland, 2008.

[95] F. Valente. Multi-stream speech recognition based on Dempster-Shafer combination rule. *Speech Communication*, 52(3):213 – 222, 2010.

[96] H. Misra. *Multi-stream processing for noise robust speech recognition*. PhD thesis, École Polytechnique Fédérale de Lausanne, 2005.

[97] H. Misra, H. Bourlard, and V. Tyagi. New entropy based combination rules in HMM/ANN multi-stream ASR. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 2, pages 741 – 744, Hong Kong, Hong Kong, 2003.

[98] J. Morris and E. Fosler-Lussier. Conditional random fields for integrating local discriminative classifiers. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 16(3):617–628, 2008.

[99] Jonathan G. Fiscus. A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER). In *Proc. of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 347–354, Santa Barbara, CA, 1997.

[100] S. Wu, B. Kingsbury, N. Morgan, and S. Greenberg. Incorporating information from syllable-length time scales into automatic speech recogniton. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 721 – 724, Seattle, WA, 1998.

[101] S. Wu, B. Kingsbury, N. Morgan, and S. Greenberg. Performance improvements through combining phone- and syllable-scale information in automatic speech recognition. In *Proc. International Conference on Spoken Language Processing (ICSLP)*, pages 459 – 462, Sidney, Australia, 1998.

[102] Katrin Kirchhoff and Jeff A. Bilmes. Combination and joint training of acoustic classifiers for speech recognition. In *Proc. ISCA ITRW Workshop on Automatic Speech Recognition*, pages 17 – 23, Paris, France, 2000.

[103] Katrin Kirchhoff, Gernot A. Fink, and Gerhard Sagerer. Conversational speech recognition using acoustic and articulatory input. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1435–1438, Istanbul, Turkey, 2000.

[104] Daniel P. W. Ellis. Stream combination before and/or after the acoustic model. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1635–1638, 2000.

[105] M. Wölmer, F. Weniger, J. Geiger, B. Schuller, and G Rigoll. Noise robust ASR in reverberated multisource environments applying convolutive NMF and long short-term memory. *Computer Speech and Language*, 27(3):780–797, 2012.

[106] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, Inc., San Francisco, CA, USA, 1988.

[107] Y. Sun, J.F. Gemmeke, B. Cranen, L. ten Bosch, and L. Boves. Using a DBN to integrate sparse classification and GMM-based ASR. In *Proc. of the International Speech Communication Association (INTERSPEECH)*, Makuhari, Japan, 2010.

[108] Yang Sun, Jort F. Gemmeke, Bert Cranen, Louis ten Bosch, and Lou Boves. Improvements of a dual-input DBN for noise robust ASR. In *Proc. of the International Speech Communication Association (INTERSPEECH)*, pages 1669 – 1672, Makuhari, Japan, 2011.

[109] J. Bilmes. The GMTK documentation, 2002.

[110] Jeff Bilmes. Graphical models and automatic speech recognition. Technical Report UWEETR-2001-0005, University of Washington, Department of Electrical Engineering, Seattle, WA, 2001.

[111] Özgür Çetin. *Multi-rate modeling, model inference, and estimation for statistical classifiers*. PhD thesis, University of Washington, Seattle, WA, 2005.

[112] Kate Saenko, Karen Livescu, James Glass, and Trevor Darrell. Multistream articulatory feature-based models for visual speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31:1700–1707, 2009.

[113] J. Bilmes, G. Zweig, T. Richardson, K. Filali, K. Livescu, P. Xu, K. Jackson, Y. Brandman, E. Sandness, E. Holtz, J. Torres, and B. Byrne. Discriminatively structured graphical models for speech recognition: JHU-WS-2001 final workshop report. Technical report, CLSP, Johns Hopkins University, Baltimore MD, 2001.

[114] A. Subramanya, C. Bartels, J. Bilmes, and P. Nguyen. Uncertainty in training large vocabulary speech recognizers. In *Proc. of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Kyoto, Japan, 2007.

[115] J. Bilmes. On virtual evidence and soft evidence in Bayesian networks. Technical Report UWEETR-2004-0016, University of Washington, Dept. of Electrical Engineering, Seattle, WA, 2004.

[116] Chia-Ping Chen and Jeff A. Bilmes. MVA processing of speech features. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 15(1):257–270, 2007.

[117] Jort F. Gemmeke, Antti Hurmalainen, Tuomas Virtanen, and Yang Sun. Toward a practical implementation of exemplar-based noise robust ASR. In *Proc. of European Signal Processing Conference (EUSIPCO)*, pages 1490–1494, Barcelona, Spain, 2011.

[118] Jort F. Gemmeke and Hugo Van hamme. A hierarchical exemplar-based sparse model of speech, with an application to ASR. In *Proc. of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Hawai, USA, 2011.

[119] A. Hurmalainen, K. Mahkonen, J. F. Gemmeke, and T. Virtanen. Exemplar-based recognition of speech in highly variable noise. In *Proc. International Workshop on Machine Listening in Multisource Environments*, Florence, Italy, 2011.

[120] Hesham Tolba, Sid-Ahmed Selouani, and Douglas D. O'Shaughnessy. Auditory-based acoustic distinctive features and spectral cues for automatic speech recognition using a multi-stream paradigm. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 837–840, 2002.

[121] András Zolnay, Daniil Kocharov, Ralf Schlüter, and Hermann Ney. Using multiple acoustic feature sets for speech recognition. *Speech Communication*, 49(6):514–525, 2007.

[122] Ralf Schlüter and Hermann Ney. Using phase spectrum information for improved speech recognition performance. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 133–136, Salt Lake City, Utah, 2001.

[123] Andras Zolnay, Ralf Schlüter, and Hermann Ney. Acoustic feature combination for robust speech recognition. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 1, pages 457–460, Philadelphia, PA, 2005.

[124] Hervé Bourlard and Sthéhane Dupont. A mew ASR approach based on independent processing and recombination of partial frequency bands. In *Proceedings of ICSLP-96*, volume 1, pages 426–429, 1996.

[125] H Hennansky, Sangita Tibrewala, and Misha Pave. Towards ASR on partially corrupted speech. In *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*, volume 1, pages 462–465, 1996.

[126] Peter Beyerlein. Discriminative model combination. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 1, pages 481–484, 1998.

[127] Xiang Li, Rita Singh, and Richard M. Stern. Combining search spaces of heterogeneous recognizers for improved speech recogniton. In *Proc. of the International Speech Communication Association (INTERSPEECH)*, 2002.

[128] Rita Singh, Michael L Seltzer, Bhiksha Raj, and Richard M Stern. Speech in noisy environments: robust automatic segmentation, feature extraction, and hypothesis combination. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 1, pages 273–276, 2001.

[129] Dimitra Vergyri. *Integration of multiple knowledge sources in speech recognition using minimum error training.* PhD thesis, The Johns Hopkins University, 2001.

[130] Katrin Kirchhoff and Jeff A. Bilmes. Dynamic classifier combination in hybrid speech recognition systems using utterance-level confidence values. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1999.

[131] C. Genest and J. V. Zidek. Combining probability distributions: A critique and an annotated bibliography. *Statistical Science*, 1(1):114–135, 1986.

[132] Josef Kittler, Mohamad Hatef, Robert P. W. Duin, and Jiri Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998.

[133] David MJ Tax, Martijn Van Breukelen, Robert PW Duin, and Josef Kittler. Combining multiple classifiers by averaging or by multiplying? *Pattern recognition*, 33(9):1475–1485, 2000.

[134] Herve A. Bourlard and Nelson Morgan. *Connectionist Speech Recognition: A Hybrid Approach*. Kluwer Academic Publishers, Norwell, MA, USA, 1993.

[135] Hemant Misra. *Multi-stream processing for noise robust speech recognition*. PhD thesis, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, 2006. IDIAP-RR 2006 28.

[136] Y. Sun, J.F. Gemmeke, B. Cranen, L. ten Bosch, and L. Boves. Using sparse representations for exemplar based continuous digit recognition. In *Proc. of European Signal Processing Conference (EUSIPCO)*, Barcelona, Spain, 2011.

[137] N. Morgan and Hervé Bourlard. Continuous speech recognition: An introduction to the hybrid HMM/connectionist approach. *IEEE Signal processing magazine*, 12(3):25–42, 1995.

[138] David Johnson, Dan Ellis, Chris Oei, Chuck Wooters, Philipp Faerber, N Morgan, and K Asanovic. ICSI Quicknet software package, 2004.

[139] Y. Sun, J. Gemmeke, B. Cranen, L. ten Bosch, and L. Boves. Fusion of parametric and non-parametric approaches to noise-robust ASR. *Speech Communication*, 56:49–62, 2014.

[140] H. Bourlard and C.J. Wellekens. Links between Markov models and multilayer perceptrons. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(12):1167–1178, 1990.

[141] H. Bourlard and N. Morgan. *Connectionist speech recognition: A hybrid approach*. Kluwer, 1994.

[142] Sree Hari Krishnan Parthasarathi, Mathew Magimai-Doss, Herv Bourlard, and Daniel Gatica-Perez. Evaluating the robustness of privacy-sensitive audio features for speech detection in personal audio log scenarios. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4474–4477, 2010.

[143] Benjamin Picart. Improved phone posterior estimation through k-NN and MLP-based similarity. Idiap-RR Idiap-RR-18-2009, Idiap, 2009.

[144] Afsaneh Asaei, Benjamin Picart, and Hervé Bourlard. Analysis of phone posterior feature space exploiting class-specific sparsity and MLP-based similarity measure. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4886–4889, 2010.

[145] George E Dahl, Dong Yu, Li Deng, and Alex Acero. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 20(1):30–42, 2011.

[146] Geoffrey Hinton, Li Deng, Dong Yu, George Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Brian Kingsbury, et al. Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal processing magazine*, 29, 2012.

[147] W. Xiong, L. Wu, F. Alleva, J. Droppo, X. Huang, and A. Stolcke. The Microsoft 2017 Conversational Speech Recognition System. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5934 – 5938, 2018.

[148] Raphael Tang and Jimmy Lin. Deep residual learning for small-footprint keyword spotting. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5484 – 5488, 2018.

[149] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural networks*, 18(5-6):602–610, 2005.

[150] Felix A Gers, Nicol N Schraudolph, and Jürgen Schmidhuber. Learning precise timing with LSTM recurrent networks. *Journal of machine learning research*, 3(Aug):115–143, 2002.

[151] Haşim Sak, Andrew Senior, and Françoise Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *Proc. of the International Speech Communication Association (INTERSPEECH)*, 2014.

[152] Andreas Stolcke, Yochai Konig, and Mitchel Weintraub. Explicit word error minimization in N-best list rescoring. In *Proc. of European Conference on Speech Communication and Technology*, 1997.

[153] Haihua Xu, Daniel Povey, Lidia Mangu, and Jie Zhu. An improved consensus-like method for minimum Bayes risk decoding and lattice combination. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4938–4941, 2010.

[154] Kornel Laskowski, Mattias Heldner, and Jens Edlund. The fundamental frequency variation spectrum. In *Proc. of FONETIK*, pages 29–32, Gothenburg, Sweden, 2008. Citeseer.

[155] Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet. Frontend factor analysis for speaker verification. *IEEE Transactions on Acoustics, Speech and Language Processing*, 19(4):788–798, 2011.

[156] Jonathan Fiscus. A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER). *IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, 2000.

[157] Lidia Mangu, Eric Brill, and Andreas Stolcke. Finding consensus in speech recognition: word error minimization and other applications of confusion networks. *Computer Speech & Language*, 14(4):373 – 400, 2000.

[158] Fang-Kuei Li. Languages and dialects of China. *Journal of Chinese Linguistics*, pages 1–13, 1973.

[159] Longsheng Guo. The relationship between Putonghua and Chinese dialects. In *Language policy in the People's Republic of China*, pages 45–54. Springer, 2004.

[160] Stephen A. Wurm, Rong Li, Theo Baumann, and Mei W. Lee. *Language Atlas of China*. Longman, 1987.

[161] Jerry Norman. *Chinese*. Cambridge University Press, Cambridge, UK, 1988.

[162] Chao Huang, Tao Chen, and Eric Chang. Accent issues in large vocabulary continuous speech recognition. *International Journal of Speech Technology*, 7 (2-3):141–153, 2004.

[163] Dong Yang, Iwano Koji, and Sadaoki Furui. Accent analysis for Mandarin large vocabulary continuous speech recognition. Technical report, Institute of Electronics, Information and Communication Engineers, Tokyo, Japan, 2008.

[164] Felix Weninger, Yang Sun, Junho Park, Daniel Willett, and Puming Zhan. Deep learning based Mandarin accent identification for accent robust ASR. In *Proc. of the International Speech Communication Association (INTERSPEECH)*, pages 510–514, 2019.

[165] George Saon, Hagen Soltau, David Nahamoo, and Michael Picheny. Speaker adaptation of neural network acoustic models using i-vectors. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 55–59, 2013.

[166] Vassilios V Digalakis, Dimitry Rtischev, and Leonardo G Neumeyer. Speaker adaptation using constrained estimation of gaussian mixtures. *IEEE Transactions on Speech and Audio Processing*, 3(5):357–366, 1995.

[167] Mark JF Gales. Maximum likelihood linear transformations for HMM-based speech recognition. *Computer Speech & Language*, 12(2):75–98, 1998.

[168] Jason J Humphries and Philip C Woodland. The use of accent-specific pronunciation dictionaries in acoustic model training. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 1, pages 317–320, 1998.

[169] Mingkuan Liu, Bo Xu, Taiyi Hunng, Yonggang Deng, and Chengrong Li. Mandarin accent adaptation based on context-independent/context-dependent pronunciation modeling. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 2, pages II1025–II1028, 2000.

[170] Chao Huang, Eric Chang, Jianlai Zhou, and Kai-Fu Lee. Accent modeling based on pronunciation dictionary adaptation for large vocabulary Mandarin speech recognition. In *Sixth International Conference on Spoken Language Processing*, 2000.

[171] Zongge Li, E Chong Tan, I McLoughlin, and TT Teo. Proposal of standards for intelligibility tests of Chinese speech. *IEE Proceedings-Vision, Image and Signal Processing*, 147(3):254–260, 2000.

[172] Ananthanarayan Krishnan, Yisheng Xu, Jackson T Gandour, and Peter A Cariani. Human frequency-following response: representation of pitch contours in Chinese tones. *Hearing research*, 189(1-2):1–12, 2004.

[173] S. Duanmu. *The Phonology of Standard Chinese*. Oxford linguistics. Oxford University Press, 2002.

[174] Yuen Ren Chao. *Mandarin primer: An intensive course in spoken Chinese*. Harvard University Press, 1948.

[175] Chao Yuen Ren. *A grammar of spoken Chinese*. University of California Press, Berkeley, Ca., 1968.

[176] Mingming Chen, Zhanlei Yang, Jizhong Liang, Yanpeng Li, and Wenju Liu. Improving deep neural networks based multi-accent Mandarin speech recognition using i-vectors and accent-specific top layer. In *Proc. of the International Speech Communication Association (INTERSPEECH)*, pages 3620–3624, Dresden, Germany, 2015.

[177] Jiangyan Yi, Hao Ni, Zhengqi Wen, and Jianhua Tao. Improving BLSTM RNN based Mandarin speech recognition using accent dependent bottleneck features. In *Proc. of Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA-ASC)*, pages 1–5, Jeju, Korea, 2016.

[178] Najim Dehak, Pedro A Torres-Carrasquillo, Douglas Reynolds, and Reda Dehak. Language recognition via ivectors and dimensionality reduction. In *Proc. of the International Speech Communication Association (INTERSPEECH)*, pages 857–860, Florence, Italy, 2011.

[179] Björn W Schuller, Stefan Steidl, Anton Batliner, Julia Hirschberg, Judee K Burgoon, Alice Baird, Aaron C Elkins, Yue Zhang, Eduardo Coutinho, and

Keelan Evanini. The INTERSPEECH 2016 Computational Paralinguistics Challenge: Deception, Sincerity & Native Language. In *Proc. of the International Speech Communication Association (INTERSPEECH)*, pages 2001–2005, San Francisco, CA, 2016.

[180] Alberto Abad, Eugénio Ribeiro, Fábio Kepler, Ramón Fernández Astudillo, and Isabel Trancoso. Exploiting phone log-likelihood ratio features for the detection of the native language of non-native english speakers. In *Proc. of the International Speech Communication Association (INTERSPEECH)*, pages 2413–2417, San Francisco, CA, 2016.

[181] Mohammed Senoussaoui, Patrick Cardinal, Najim Dehak, and Alessandro L Koerich. Native language detection using the i-vector framework. In *Proc. of the International Speech Communication Association (INTERSPEECH)*, pages 2398–2402, San Francisco, CA, 2016.

[182] Xuesong Yang, Kartik Audhkhasi, Andrew Rosenberg, Samuel Thomas, Bhuvana Ramabhadran, and Mark Hasegawa-Johnson. Joint modeling of accents and acoustics for multi-accent speech recognition. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, 2018.

[183] Yi Liu and Pascale Fung. Multi-accent Chinese speech recognition. In *Proc. of International Conference on Spoken Language Processing (ICSLP)*, pages 133–136, Pittsburgh, PA, 2006.

[184] Tingyao Wu, Jacques Duchateau, Jean-Pierre Martens, and Dirk Van Compernolle. Feature subset selection for improved native accent identification. *Speech Communication*, 52(2):83 – 98, 2010.

[185] Charles Chen, Razvan C Bunescu, Li Xu, and Chang Liu. Tone classification in Mandarin Chinese using convolutional neural networks. In *Proc. of the International Speech Communication Association (INTERSPEECH)*, pages 2150–2154, San Francisco, CA, 2016.

[186] Gina-Anne Levow. Context in multi-lingual tone and pitch accent recognition. In *Proc. of European Conference on Speech Communication and Technology*, pages 1809–1812, Lisbon, Portugal, 2005.

[187] Li Zhang, Chao Huang, Min Chu, Frank Soong, Xianda Zhang, and Yudong Chen. Automatic detection of tone mispronunciation in Mandarin. In *Proc.*

of International Symposium on Chinese Spoken Language Processing, volume 4274 of *Lecture Notes in Computer Science*, pages 590–601. Springer, 2006.

[188] Too Chen, Chao Huang, Eric Chang, and Jingehan Wang. Automatic accent identification using Gaussian mixture models. In *Proc. of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 343–346, Madonna di Campiglio, Italy, 2001.

[189] Yanli Zheng, Richard Sproat, Liang Gu, Izhak Shafran, Haolang Zhou, Yi Su, Daniel Jurafsky, Rebecca Starr, and Su-Youn Yoon. Accent detection and speech recognition for Shanghai-accented Mandarin. In *Proc. of the International Speech Communication Association (INTERSPEECH)*, pages 217–220, 2005.

[190] Yuan-Fu Liao, Shuan-Chen Yeh, Ming-Feng Tsai, Wei-Hsiung Ting, and Sen-Chia Chang. Latent prosody model-assisted Mandarin accent identification. In *Proc. of 21st Conference on Computational Linguistics and Speech Processing*, pages 125–136, Taichung, Taiwan, 2009. The Association for Computational Linguistics and Chinese Language Processing (ACLCLP).

[191] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6645–6649, Vancouver, Canada, 2013.

[192] George Saon, Hagen Soltau, David Nahamoo, and Michael Picheny. Speaker adaptation of neural network acoustic models using i-vectors. In *Proc. of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 55–59, Olomouc, Czech Republic, 2013.

[193] Felix Weninger, Erik Marchi, and Björn Schuller. Improving recognition of speaker states and traits by cumulative evidence: Intoxication, sleepiness, age and gender. In *Proc. of the International Speech Communication Association (INTERSPEECH)*, pages 1159–1162, Portland, OR, 2012.

[194] Kai Chen and Qiang Huo. Training deep bidirectional LSTM acoustic model for LVCSR by a context-sensitive-chunk BPTT approach. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 24(7): 1185–1193, 2016.

[195] Abdel-rahman Mohamed, Frank Seide, Dong Yu, Jasha Droppo, Andreas Stoicke, Geoffrey Zweig, and Gerald Penn. Deep bi-directional recurrent networks over spectral windows. In *Proc. of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 78–83, Scottsdale, AZ, 2015.

[196] Joseph B Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, 1964.

[197] Xinwei Li, Yue Pan, Matthew Gibson, and Puming Zhan. DNN online adaptation for automatic speech recognition. In *Proc. of 29th Conference on Electronic Speech Signal Processing (ESSV)*, Ulm, Germany, 2018.

[198] Richard P Lippmann. Speech recognition by machines and humans. *Speech communication*, 22(1):1–15, 1997.

[199] Harlan Lane and Bernard Tranel. The Lombard sign and the role of hearing in speech. *Journal of Speech and Hearing Research*, 14(4):677–709, 1971.

[200] Lynn K Perry, Emily N Mech, Maryellen C MacDonald, and Mark S Seidenberg. Influences of speech familiarity on immediate perception and final comprehension. *Psychonomic Bulletin & Review*, 25(1):431–439, 2018.

[201] Wayne Xiong, Jasha Droppo, Xuedong Huang, Frank Seide, Mike Seltzer, Andreas Stolcke, Dong Yu, and Geoffrey Zweig. Achieving human parity in conversational speech recognition. *arXiv preprint arXiv:1610.05256*, 2016.

[202] Constantin Spille, Birger Kollmeier, and Bernd T Meyer. Comparing human and automatic speech recognition in simple and complex acoustic scenes. *Computer Speech & Language*, 52:123–140, 2018.

[203] Thai-Son Nguyen, Sebastian Stueker, and Alex Waibel. Super-human performance in online low-latency recognition of conversational speech. *arXiv preprint arXiv:2010.03449*, 2020.

[204] E Colin Cherry. Some experiments on the recognition of speech, with one and with two ears. *The Journal of the acoustical society of America*, 25(5): 975–979, 1953.

[205] Donald Eric Broadbent. *Perception and communication*. Elsevier, 2013.

[206] Quentin Summerfield. Audio-visual speech perception, lipreading and artificial stimulation. In *Hearing science and hearing disorders*, pages 131–182. Elsevier, 1983.

[207] Louis CW Pols. Flexible human speech recognition. In *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pages 273–283, 1997.

[208] Ernst F Schröder. Voice control system with a microphone array, 2005. US Patent 6,868,045.

[209] Stefan Braun, Daniel Neil, Jithendar Anumula, Enea Ceolini, and Shih-Chii Liu. Multi-channel attention for end-to-end speech recognition. In *Proc. of the International Speech Communication Association (INTERSPEECH)*. Proceedings of Interspeech 2018, 2018.

[210] Paul Duchnowski, Uwe Meier, and Alex Waibel. See me, hear me: Integrating automatic speech recognition and lip-reading. In *Third International Conference on Spoken Language Processing*, 1994.

[211] Stéphane Dupont and Juergen Luettin. Audio-visual speech modeling for continuous speech recognition. *IEEE Transactions on Multimedia*, 2(3): 141–151, 2000.

[212] Jing Huang and Brian Kingsbury. Audio-visual deep learning for noise robust speech recognition. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7596–7599, 2013.

[213] Kuniaki Noda, Yuki Yamaguchi, Kazuhiro Nakadai, Hiroshi G Okuno, and Tetsuya Ogata. Audio-visual speech recognition using deep learning. *Applied Intelligence*, 42(4):722–737, 2015.

[214] Zoltán Tüske, Wilfried Michel, Ralf Schlüter, and Hermann Ney. Parallel neural network features for improved tandem acoustic modeling. In *Proc. of the International Speech Communication Association (INTERSPEECH)*, pages 1651–1655, 2017.

[215] Jinyu Li, Li Deng, Yifan Gong, and Reinhold Haeb-Umbach. An overview of noise-robust automatic speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(4):745–777, 2014.

[216] Eva Reinisch. Speaker-specific processing and local context information: The case of speaking rate. *Applied Psycholinguistics*, 37(6):1397–1415, 2016.

[217] Xiaofei Wang, Ruizhi Li, and Hynek Hermansky. Stream attention for distributed multi-microphone speech recognition. In *Proc. of the International Speech Communication Association (INTERSPEECH)*, pages 3033–3037, 2018.

[218] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[219] Pengfei Liu, Kun Li, and Helen Meng. Group gated fusion on attention-based bidirectional alignment for multimodal emotion recognition. In *Proc. of the International Speech Communication Association (INTERSPEECH)*, pages 379–383, 2020.

[220] Jing Pan, Joshua Shapiro, Jeremy Wohlwend, Kyu J Han, Tao Lei, and Tao Ma. ASAPP-ASR: Multistream CNN and self-attentive SRU for SOTA speech recognition. *arXiv preprint arXiv:2005.10469*, 2020.

[221] Suyoun Kim, Takaaki Hori, and Shinji Watanabe. Joint CTC-attention based end-to-end speech recognition using multi-task learning. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4835–4839, 2017.

[222] Shinji Watanabe, Takaaki Hori, Suyoun Kim, John R Hershey, and Tomoki Hayashi. Hybrid CTC/attention architecture for end-to-end speech recognition. *IEEE Journal of Selected Topics in Signal Processing*, 11(8):1240–1253, 2017.

[223] Takaaki Hori, Shinji Watanabe, Yu Zhang, and William Chan. Advances in joint CTC-attention based end-to-end speech recognition with a deep CNN encoder and RNN-LM. *arXiv preprint arXiv:1706.02737*, 2017.

[224] Mark JF Gales and Philip C Woodland. Mean and variance adaptation within the MLLR framework. *Computer speech and language*, 10(4):249–264, 1996.

[225] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur. X-vectors: Robust DNN embeddings for speaker recognition. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5329–5333, 2018.

[226] MA Tuğtekin Turan, Emmanuel Vincent, and Denis Jouvet. Achieving multi-accent ASR via unsupervised acoustic model adaptation. In *Proc. of the International Speech Communication Association (INTERSPEECH)*, 2020.

[227] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[228] Tanja Schultz and Alex Waibel. Language-independent and language-adaptive acoustic modeling for speech recognition. *Speech Communication*, 35(1-2): 31–51, 2001.

[229] Ramya Rasipuram and Mathew Magimai-Doss. Acoustic and lexical resource constrained ASR using language-independent acoustic model and language-dependent probabilistic lexical model. *Speech Communication*, 68:23–40, 2015.

[230] William Byrne, Peter Beyerlein, Juan M Huerta, Sanjeev Khudanpur, Bhaskara Marthi, John Morgan, Nino Peterek, Joe Picone, Dimitra Vergyri, and T Wang. Towards language independent acoustic modeling. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 2, pages 1029–1032, 2000.

[231] Yang Sun, Junho Park, WEI Goujin, and Daniel Willett. System and method for accent classification, March 18 2021. US Patent App. 16/570,122.

Summary

Chapter 1: Introduction

Automatic speech recognition (ASR) is the process that maps speech signals to a sequence of words. While ASR has become reasonably accurate in quiet environments and with standard pronunciations, performance deteriorates dramatically in the real world where speech signals are often contaminated by various types of background noise and speakers may deviate from standard pronunciation in terms of speaking speed, tones and accents.

This PhD thesis studies both noise-robust and accent-robust ASR. To improve noise robustness, I focused on system-combination approaches to harness the strengths of different systems. For noise robustness, several combination algorithms are developed, including acoustic feature transformation, dynamic weight estimate and multi-stream combination. For accent robustness, adaptation of the pronunciation model was studied at different levels of phonetic confusions. Also, an accent classifier was built to allow more accent-specific enhancement. The main issues addressed in this thesis for noise and accent robustness can be found below:

Issues in system combination for noise-robust ASR:

1. In the real world there is a large range of noise types and signal-to-noise ratios. It is extremely difficult for an ASR system to perform well across all/most noisy conditions.

2. Most of the system combination approaches are designed for systems that use highly similar features and that have similar performance levels. It is not well known how the strong points of systems that use very different features or have different performance can be harnessed to obtain the best possible combination.

Issues in pronunciation model adaptation for accent-robust ASR:

1. Regional and socio-economic differences correlate with a wide range of different accents. It is not easy to create an accent-independent robust system.

2. A good accent classifier for Mandarin is missing. The number of established accents in Mandarin is quite large; moreover, there is substantial between-speaker variation within the individual accents.

Chapter 2: Tandem Approach on Top of an Exemplar-Based System for Noise-Robust ASR

A Gaussian Mixture Model (GMM)-based tandem system is proposed in Chapter 2 to model probability estimates of an exemplar-based system, so called Sparse Classification (SC). Two novel transformations are introduced: (i) Gaussianization of tiny probability entries alleviates the long tail problem in GMM modelling and (ii) histogram normalization mitigates the mismatch problem between training and test data across a wide range of signal-to-noise ratios (SNRs). The transformed probability estimates of the SC system are stacked on top of traditional acoustic features as the input to the GMM in a tandem combination.

The method is tested on Aurora 2 - a continuous digit recognition task, showing 53.5% and 10.93% Word Error Rate Reduction (WERR) over an MFCC-based baseline on speech with seen and unseen noise types, respectively. Subsequent experiments on Aurora 4 - a large vocabulary task with noise, yield 10.8% WERR on seen additive noise. However, WERR becomes -14.6% on test data contaminated by convolutional noises, because the SC technique is restricted to additive speech/noise representation and incapable of handling convolutional noises.

Chapter 3: Fusion of Parametric and Non-parametric Approach to Noise -Robust ASR

Rather than being transformed and modelled by GMMs, the SC probability scores are integrated into an MFCC-based GMM system in the form of Virtual Evidence (VE) using a Dynamic Bayesian Network (DBN) platform. In this way, the SC predictions become a second opinion, not only in the inference stage, but it can also play a role in the joint GMM training. As a result, the long tail of the SC estimate is no longer a problem, and adjusting the weights of the two input streams - MFCC and SC - becomes easier than in the tandem approach introduced in Chapter 2.

Experiments on Aurora 2 show that a substantial gain in performance can be obtained over the better of the two individual systems across a wide range of SNRs after a careful optimization of the weight set. That is 23.8% WERR over GMM at

clean speech and 12.9% WERR over SC at SNR -5dB on test set B that contains noises not seen in training.

Chapter 4: Multi-stream System Combination with Confidence-based Adaptive Weights for Robust ASR

Chapter 4 further improves the system combination in the probability domain. Instead of the traditional inverse entropy as the combination weights, I proposed to estimate confidence for each component stream independently, so that the combination can be applied to systems that differ substantially in the way in which they estimate the likelihood of subword units, such as Multi-Layer Perceptron (MLP) and SC. Moreover, sum- and product-based combination approaches are studied, as well as a dynamic switch between the two combination strategies.

Experiment on Aurora 2 indicates that the dynamic weighting scheme can find a weight set which stays in the optimal weight envelope. The final results are even better than the oracle static weighting scheme at utterance level, given the knowledge of SNR, by 4.4% and 8.6% on test data with seen and unseen noise types, respectively. This shows that an adaptive frame-level weighting adjustment outperforms static weights derived from oracle knowledge about the SNR at utterance level.

Chapter 5: Off-line Lattice Combination with Dynamic Weights on Large Vocabulary Continuous Speech Recognition Tasks

Chapter 5 generalizes the combination idea to large vocabulary tasks in Mandarin Chinese. First, a dedicated confidence model is built for each of five systems that can be combined. Second, lattice combination replaces probability combination. Third, the combination is extended up to five component systems.

Experiments on real-world test set show that 1) complementary information can be exploited when combining acoustic models with different architectures, but trained with the same set of data, 2) the advantage even holds for the same acoustic models with different decoding operation points (decoding parameters such as word insertion penalty) and 3) a five-way combination yields 14.9% Character Error Rate Reduction (CERR) over a baseline system that uses acoustic features based on a Feed-Forward Dynamic Neural Network.

Chapter 6: Lexicon Study towards Accent-Robust Mandarin ASR

Chapter 6 provides an accent-robustness study for Mandarin ASR by pursuing improvement of the acoustic model (AM) or the pronunciation model (PM). In the AM approach, toneless phones are introduced and the corresponding toneless data is mixed with original tonal data in a joint training. Under the PM umbrella, a data-driven PM adaptation at different sub-word unit levels is investigated. The PM approach is shown to be more effective than the AM approach, in terms of the length of the development period and it offers the possibility of a fine-grained optimization for specific pronunciation phenomena.

The proposed approaches are evaluated on a large data collection of 15 Mandarin accents. The AM approach always requires a compromise between performances on heavy and light accents. But the PM approach achieves 32.1% and 9.5% CERR on the two accent groups, respectively.

Chapter 7: Deep-Learning-based Mandarin Accent Identification for Accent-Robust ASR

Chapter 7 is an in-depth study into the classification of regional accents in Mandarin speech. Both bi-directional long short-term memory (LSTM) networks and i-vectors are investigated in an accent classifier. Non-metric dimensional scaling (NMDS) showed that the 15 Mandarin accents form three groups that can be characterized as 'standard', 'light accent' and 'heavy accent'.

With i-vector speaker adaptation, relative CERRs of 13.2%, 15.3% and 14.6% are obtained on standard, light-accented and heavy-accented Mandarin, respectively. Accent classifiers enable further accent-specific optimization of the PM approach introduced in Chapter 6.

Chapter 8: General Discussion and Concluding Remarks

In addition to summarizing the results obtained with respect to the four issues described in the Introduction, Chapter 8 discusses how the techniques proposed in this thesis can be applied beyond noise- and accent-robust ASR.

The core contributions of the system combination technique investigated in this thesis are two-fold. First, several transformations are introduced for harnessing

the strengths of multiple systems that may differ substantially in approach and performance level. Second, a dynamic weighting algorithm is developed to allow dynamic and adaptive adjustment of the importance of each component system in diverse scenarios. Although the system combination study was initiated to solve recognition deteriorations caused by background noise, the algorithms are clearly not limited to improving noise robustness. Several other scenarios in which the combination technique can be applied are discussed in Chapter 8. Like humans who do not perceive the world via one single channel, also ASR and automatic speech understanding can benefit from the fusion of multi-stream inputs.

With respect to accent-robustness, a data-driven approach was developed to adapt the lexicon, so that the recognizer becomes more tolerant to pronunciation variations associated with different accents. The adaptation approach is shown to be efficient and effective for improving accent robustness against all15 Mandarin accents at the same time. Moreover, an accent classifier was introduced to allow the lexicon adaptation to be more accent-specific. Beyond ASR, the accent classifier can be used as a user profiling approach for a better personalization in a comprehensive conversational system that includes natural language understanding, natural language generation and speech synthesis.

Samenvatting

Hoofdstuk 1. Inleiding

Automatische Spraakherkenning (ASH) vertaalt spraaksignalen naar rijen woorden in schrift. ASH is momenteel redelijk nauwkeurig als de spreker zich in stille omgevingen bevindt, en zich bedient van standaard uitspraak. Maar de nauwkeurigheid gaat hard achteruit als er achtergrondlawaai is, of als de spraak afwijkt van de 'standaard' wat spreeksnelheid of uitspraak betreft. Dit proefschrift bevat onderzoek naar ASH-technieken die bestand zijn tegen achtergrondlawaai en sprekers met een regionaal accent. Pogingen om de robuustheid tegen achtergrondlawaai te verbeteren zijn gebaseerd op de combinatie van verschillende ASH-technieken, om te profiteren van de sterke kanten van die technieken. In dat kader wordt een aantal specifieke technieken ontwikkeld en getest, zoals transformaties van akoestische features en technieken om dynamische schattingen te maken van de betrouwbaarheid van de verschillende technieken. Pogingen om de robuustheid tegen accentverschillen te verbeteren richten zich vooral op de aanpassing en uitbreiding van de fonetische transcripties van de woorden in het lexicon. Daarbij worden aanpassingen op verschillende niveaus van detail in de transcriptie van woorden in het Mandarijn Chinees vergeleken. Ook wordt een techniek ontwikkeld om de 15 belangrijkste accenten in het Mandarijn te herkennen, waardoor de aanpassingen in het lexicon specifieker kunnen zijn.

Meer in detail richt het onderzoek naar robuustheid tegen achtergrondlawaai zich op twee problemen:

1. De variatie in soorten achtergrondlawaai en in het niveau van dat lawaai is in de 'echte wereld' bijzonder groot. Dat maakt het lastig om een ASH-techniek te ontwikkelen die in bijna al die verschillende omstandigheden goed presteert.

2. De meeste technieken voor het combineren van systemen die in de literatuur beschreven zijn richten zich op systemen die alleen in details van elkaar verschillen en die ook qua nauwkeurigheid dicht in elkaars buurt liggen. Hier wordt onderzocht hoe technieken die sterk van elkaar verschillen, en die qua nauwkeurigheid in sommige situaties sterk van elkaar kunnen verschillen toch met elkaar gecombineerd kunnen worden, zodat de techniek die in die situatie het beste is het grootste gewicht krijgt.

Ook het onderzoek naar robuustheid tegen accentverschillen in ASH richt zich op twee problemen:

1. Verschillen in uitspraak die samenhangen met regionale achtergrond en sociaaleconomische status van de sprekers zijn enorm groot. Het is lastig om een techniek te ontwikkelen die voor alle accenten even nauwkeurig werkt.

2. Er bestaat nog geen systeem dat de 15 belangrijkste accenten in het Mandarijn Chinees kan herkennen. Daar komt bij dat de verschillen tussen sprekers van hetzelfde accent groot kunnen zijn.

Hoofdstuk 2. Verbetering van ASH door de combinatie van een systeem gebaseerd op exemplars en een systeem gebaseerd op Gaussian Mixtures

De kansverdelingen van de akoestische features die geleverd worden door een systeem dat exemplars gebruikt in een sparse classification techniek verschillen sterk van de verdelingen van de features in conventionele ASH systemen. In dit hoofdstuk worden twee manieren onderzocht om de zeer scheve verdelingen van de sparse classification techniek te transformeren naar een meer symmetrische vorm. (1) Het vervangen van de heel kleine kansen in de lange staart van de verdelingen maakt het makkelijker om Gaussian Mixture Models te trainen. (2) Histogram Normalisering verkleint de mismatch tussen de verdelingen in training- en testdata over een breed scala van signaal-ruisverhoudingen. Die getransformeerde features worden vervolgens samen met conventionele features gebruikt als input voor een model gebaseerd op Gaussian Mixtures. De aanpak wordt getest door experimenten met Aurora-2, een database met sequenties van de cijfers nul tot en met negen. Het tandem systeem leidt tot een reductie van 53.55% van de fouten voor hetzelfde achtergrondlawaai als waarmee de herkenner getraind is. Voor onbekend achtergrondlawaai is de reductie van het aantal fouten 10.93%. De aanpak wort vervolgens getest op Aurora-4, een database met voorgelezen spraak. Hier is de reductie van fouten 10.8%. Een laatste experiment, ook met Aurora-4, maar nu met andere vervormingen van de spraaksignalen zoals echos, leidt tot een toename van het aantal fouten met 14.6%. Dit kan verklaard worden door het gegeven dat sparse classification alleen kan omgaan met achtergrondlawaai dat bij het signaal opgeteld wordt, zonder dat signaal ook op andere manieren te vervormen.

Hoofdstuk 3. Fusie van parametrische en non-parametrische benaderingen voor het verbeteren van de robuustheid tegen achtergrondlawaai

Dit hoofdstuk beschrijft een systeem waarin de waarschijnlijkheidsscores van een sparse classification systeem gefuseerd worden met corresponderende scores van een conventioneel ASH systeem door gebruik te maken van het concept 'virtual evidence' in dynamische Bayesiaanse netwerken. In deze opzet vormen de scores van het sparse classification systeem een soort 'second opinion' en is de scheefheid van de waarschijnlijkheidsverdelingen in het sparse classification systeem niet langer een probleem. In deze aanpak kunnen de scores van de te combineren systemen al in de trainingsfase samen gebruikt worden. Bovendien wordt het gemakkelijker om de dynamische gewichten die aan de evidentie van de twee systemen toegekend wordt te optimaliseren. Experimenten met Aurora-2 laten zien dat een substantiële vermindering van het aantal fouten verkregen wordt als de optimale gewichten gebruikt worden.

Hoofdstuk 4. Fusie van systemen met gebruik van gewichten die gebaseerd zijn op de betrouwbaarheid van de schattingen

Dit hoofdstuk beschrijft een volgende stap in de verbetering van de combinatie van waarschijnlijkheden in de output van twee verschillende systemen (een gebaseerd op multi-layer perceptrons en een ander gebaseerd op sparse classification). In plaats van de inverse van de entropie van de verdelingen, die in de literatuur gebruikt wordt om de gewichten te bepalen, wordt een manier ontwikkeld om de betrouwbaarheid van de schattingen van twee systemen te bepalen; vervolgens wordt die betrouwbaarheid gebruikt als gewicht in de fusie. Bovendien worden twee methoden voor het fuseren vergeleken, namelijk door vermenigvuldigen of door optellen van kansen. Experimenten met Aurora-2 laten zien dat deze aanpak gewichten oplevert die vrijwel optimaal zijn in een brede range van condities (type achtergrondlawaai en signaal-ruisverhouding). De resultaten met op betrouwbaarheid gebaseerde gewichten zijn zelfs beter dan de beste resultaten die behaald kunnen worden als gebruik gemaakt wordt van voorkennis over de signaal-ruisverhouding. Met hetzelfde achtergrondlawaai als in de training is de verbetering 4.4%; met achtergrondlawaai dat niet gebruikt is in de training is de verbetering 8.6

Hoofdstuk 5. Dynamische gewichten in de fusie van lattices in een ASH taak met een groot lexicon

Dit hoofdstuk generaliseert de resultaten van de experimenten in de voorafgaande hoofdstukken, en breidt die uit tot de fusie van lattices, in plaats van waarschijnlijkheden van (sub-)woord eenheden. Voor dit doel wordt allereerst een manier ontwikkeld om de kwaliteit van ASH systemen voor het Mandarijn Chinees te bepalen. Bovendien worden hier niet twee, maar vijf systemen gecombineerd. De experimenten zijn gebaseerd op spraak die opgenomen is in rijdende auto's. Het blijkt mogelijk om complementaire informatie te onttrekken aan de output van systemen die qua architectuur verschillen, maar wel met hetzelfde spraakmateriaal getraind zijn. Zelfs de combinatie van twee versies van hetzelfde systeem, maar met verschillende instellingen van controleparameters leidt tot een verkleining van het percentage foute karakters in de output. De combinatie van vijf systemen leidt tot een verlaging van het aantal fout herkende karakters van 14.9%, vergeleken met een referentiesysteem dat gebruik maakt van een feed-forward dynamische neuraal netwerk.

Hoofdstuk 6. Verbetering van ASH voor Mandarijn Chinees door aanpassing van het lexicon

Robuustheid tegen variatie in accent kan -in principe- verkregen worden door aanpassing van de akoestische modellen en/of door aanpassingen van de fonetische representaties van de woorden in het lexicon, ook wel aangeduid als een uitspraakmodel (PM). In het Mandarijn zijn er vier tonen die onderscheid maken tussen woorden/karakters. Op de eerste plaats zijn nieuwe akoestische modellen getraind door in een deel van het trainingsmateriaal de tonen weg te laten. Daarnaast is een data-gedreven onderzoek gedaan om te achterhalen welke accent-specifieke fonetische variatie aan het lexicon toegevoegd moet worden om de nauwkeurigheid van de herkenner te optimaliseren. Aanpassing van het uitspraakmodel kost minder tijd dan hertraining van de akoestische modellen. Bovendien biedt aanpassing van het uitspraakmodel veel meer mogelijkheden om specifieke problemen met afzonderlijke accenten aan te pakken. Experimenten met een grote database van Mandarijn Chinees waarin 15 regionale accenten vertegenwoordigd zijn laten zien dat de aanpak op basis van het uitspraakmodel de voorkeur verdient boven de hertraining van de akoestische modellen. Bij die hertraining moet altijd een compromis gevonden worden tussen lichte en zware accenten. Met aanpassing van het

uitspraakmodel wordt het aantal fout herkende karakters verkleind met 32.1% in de zware, en met 9.5% voor de lichte accenten.

Hoofdstuk 7. Een benadering gebaseerd op Deep-Learning voor de herkenning van accenten in het Mandarijn Chinees

Aanpassing van het uitspraakmodel is veel effectiever als het accent van een spreker bekend is. In dit hoofdstuk wordt een techniek ontwikkeld om sprekernormalisatie en accentherkenning te combineren door gebruik te maken van zogenaamde i-vectoren en bi-directionele long short-term memory netwerken. De 15 belangrijkste accenten in het Mandarijn Chinees worden geclusterd in drie groepen (standaard, licht accent, zwaar accent), met behulp van multi-dimensional scaling. Sprekernormalisatie met i-vectoren gecombineerd met accentherkenning leidt tot verbeteringen van het percentage fout herkende karakters met 31.2% bij uitspraak die dicht bij de standaard ligt, 15.3% bij lichte accenten en 14.6% bij zware accenten.

Hoofdstuk 8. Discussie en conclusies

Dit hoofdstuk begint met een samenvatting van de belangrijkste resultaten van de experimenten die in de voorafgaande hoofdstukken beschreven zijn. Het onderzoek naar robuustheid tegen lawaai heeft twee belangrijke resultaten opgeleverd. Op de eerste plaats is dat een verzameling transformaties die kansen berekend door sterk van elkaar verschillende systemen kan omzetten in een vorm die geschikt is voor combinatie met andersoortige systemen. Op de tweede plaats zijn methoden ontwikkeld voor het bepalen van de gewichten die aan verschillende informatiestromen toegekend moeten worden als die stromen gecombineerd worden. Hoewel die hier ontwikkelde transformaties en methoden in dit proefschrift alleen getest zijn in het kader van ASH mag aangenomen worden dat zij generaliseren naar allerlei andere scenario's waarin verschillende informatiestromen gecombineerd moeten (of kunnen) worden. Met betrekking tot robuustheid van ASH tegen verschillende accenten is aangetoond dat een data-gedreven aanpassing van de fonetische representaties in het lexicon de voorkeur verdient boven het accent-ongevoelig maken van de akoestische modellen. De hier ontwikkelde benadering kan overweg met alle 15 accenten in het Mandarijn Chinees. Het gebruik van een accentherkenner verbetert de nauwkeurigheid van een ASH systeem voor het Mandarijn Chinees aanzienlijk. De aanpak die ten grondslag ligt aan het systeem voor accentherkenning kan ook gebruikt worden voor het profileren van sprekers in een ruimer kader. Dat biedt

mogelijkheden voor het personaliseren van mens-machine interactie op het vlak van het begrijpen van wat met uitingen bedoeld wordt, en het genereren van gesproken output.

Curriculum Vitae

[**Professional Profile**]

Yang Sun is a team lead and principle engineer in the area of automatic speech recognition with over 8 years of experience on research, development and implementation of ML solutions, requiring solid understanding of statistics, deep learning algorithms, software development principles and project management. Experienced in global collaborations with enormous cultural diversity.

[**Experience**]

- Team Lead and Principle Engineer, Huawei. 2020.11 - Present
  - Lead the R&D of Speech&Video team for Huawei Search Engine.
  - Products: Speech Recognition, Synthesis, Translation, Video Retrieval.
  - Responsible languages: Mandarin and main European languages (English, German, French, Spanish, Italian, Portuguese and Dutch).
- Senior Research Scientist, Cerence. 2019.10 - 2020.10
  - Acoustic Modelling R&D Team.
  - Develop core neural network-based ASR algorithms.
  - Contact person between global and China ASR teams.
- Senior Research Scientist, Nuance. 2013.02 - 2019.09
  - Algorithm Research Team.
  - Develop core neural network-based ASR algorithms.
  - Owner of cloud Mandarin acoustic model R&D.

[**Education**]

- Radboud University Nijmegen: PhD (expected in 2021.09)
- Technical University of Munich: Master (w/ honors)
- Shanghai Jiao Tong University: Bachelor

Publications

- [**Patent**] System and Method for Accent Classification. **Y Sun**, J Park, G Wei, D Willett. (Applied, US Patent App. 20210082402)
- [**Patent**] System and Method for Performing Automatic Speech Recognition System Parameter Adjustment via Machine Learning. D Willett, **Y Sun**, PJ Vozila, P Zhan (Granted, US Patent App. 20200043468)
- [**Journal**] Fusion of Parametric and Non-parametric Approaches to Noise-robust ASR, **Y Sun**, JF Gemmeke, B Cranen, L ten Bosch, L Boves. Speech Communication, 2014
- [**Proceedings**] Deep Learning Based Mandarin Accent Identification for Accent Robust ASR. F Weninger, **Y Sun**, J Park, D Willett, P Zhan. Proc. Interspeech 2019, Graz, Austria.
- [**Proceedings**] Analytical Assessment of Dual-stream Merging for Noise-robust ASR. L ten Bosch, B Cranen, **Y Sun**. Proc. Interspeech 2016, San Francisco, USA.
- [**Proceedings**] Using Sparse Classification Outputs as Feature Observations for Noise Robust ASR. **Y Sun**, B Cranen, JF Gemmeke, L Boves, L ten Bosch. Proc. Interspeech 2012, Portland, USA.
- [**Proceedings**] Combination of Sparse Classification and Multilayer Perceptron for Noise-robust ASR. **Y Sun**, MM Doss, JF Gemmeke, B Cranen, L ten Bosch, L Boves. Proc. Interspeech 2012, Portland, USA.
- [**Proceedings**] Improvements of a Dual-input DBN for Noise Robust ASR. **Y Sun**, JF Gemmeke, B Cranen, L ten Bosch, L Boves. Proc. Interspeech 2011, Florence, Italy.
- [**Proceedings**] Early Fusion of Sparse Classification and GMM for Noise Robust ASR. **Y Sun**, JF Gemmeke, B Cranen, L ten Bosch, L Boves. Proc. EUSIPCO 2011, Barcelona, Spain.
- [**Proceedings**] Toward a Practical Implementation of Exemplar-based Noise Robust ASR. JF Gemmeke, A Hurmalainen, T Virtanen, **Y Sun**. Proc. EUSIPCO 2011, Barcelona, Spain.
- [**Proceedings**] Hybrid HMM/BLSTM-RNN for Robust Speech Recognition. **Y Sun**, L Ten Bosch, L Boves. Proc. TSD 2010, Brno, Czech.
- [**Proceedings**] Non-negative Matrix Factorization as Noise-robust Feature Extractor for Speech Recognition. B Schuller, F Weninger, M Wöllmer, **Y Sun**, G Rigoll. Proc. ICASSP 2010, Dallas, USA.

- [**Proceedings**] Long Short-term Memory Networks for Noise Robust Speech Recognition. M Wöllmer, **Y Sun**, F Eyben, B Schuller. Proc. INTERSPEECH 2010, Makuhari, Japan.
- [**Proceedings**] Using a DBN to Integrate Sparse Classification and GMM-based ASR. **Y Sun**, J Gemmeke, B Cranen, L ten Bosch, L Boves. Proc. Interspeech 2010, Makuhari, Japan.
- [**Proceedings**] Robust In-car Spelling Recognition - a Tandem BLSTM-HMM Approach. M Wöllmer, F Eyben, B Schuller, **Y Sun**, T Moosmayr. Proc. Interspeech 2009, Brighton, UK.

Acknowledgements

First of all, my gratitude sincerely goes to three of my supervisors Lou, Bert, Louis for giving me this great opportunity to study in the great department. It takes unconventionally long for me to get the thesis done that two out of three of my supervisors have got retired... I really appreciate that you did not give up on me and still helped me after my contract with the university ends for 7 years. I can never complete it without your help.

Mathew, as my supervisor at Idiap, helped me tremendously establish the large vocabulary system in GMTK and introduced me to the system combination world. Thank you!

Jort was a senior PhD when I joined the department but acted as my daily supervisor. Thank you very much for the great mentoring, inspiration and the unique & fascinating exemplar system which becomes an essential ingredient in my combination soup! Also for the fun experience of assembling the desktop for me, which took almost as long as assembling Avengers... :D

Thank you Daniel, who was our industry partner of the SCALE project, for giving me the amazing job at Nuance afterwards. Certainly, I appreciate your contribution and approval for the work in chapters 5, 6 and 7 of this thesis which was done at the company.

I also thank all 'SCALE' PhD and Post doc fellows Afsaneh, Arnab, Cassia, Davide, Heyun, Liang, Rahil, Mauro, Youssef, Mahaboob, Serena, Zoltan, Deepu and professors in the SCALE project who walked a great journal together with me and left me behind for a couple of years. :)

Thank Heyun for on-boarding me in Nijmegen; Eric for organizing the poker nights; Bart, Steve for our 'fighting' club; Michele, Mitch, Maarten and Job for the cool-guy basketball team; Sophie and Steven again for the great accommodation and Poutine in Montreal and equal love for all the PhD and Post doc fellows in CLST who have overlap with me: Anna, Barbara, Christina, Eva, Joost, Marijn, Mira, Rahim, Susan, Vanja and Vico, including but not only due to the shared complaints (love) of our supervisors. I certainly thank our bosses too, Catia, Helmer, Henk, Nelleke, Odette and many more.

There are also a bunch of great and beautiful souls I met at Radboud, Annemijn, Andre, Iris, Lucas, Gina, Henna, Lucas, Martin, Maitta, Ming, Nessa, Stefan, Sophie, Yanjia and Ziye.

Thank my lovely colleagues Chris, Nicolas, Jianhua, Junho, Christian, Hendrik, Shen Wang, Hui Liang, Dermot, Frank, Ralf, Murat, Guojin, Jianzhong, Xiaolin at Nuance and Marcel, Maria, Raquel, Xiao, Sascha, Uday, Hubert, Jan, Qian, Tom, Dino at Cerence.

I would like to acknowledge with gratitude, the support and love of my family – first my girl Xijia who abandoned her study at LU Munich and companied me to this brand new journey. Finally I got this point to wish you could get your PhD soon. Second to my parents and grandparents. I cannot regret more that my grandfather and grandmother-in-law can never see this work got approved eventually. Procrastination really really hurts.

Special thank you for our cat Ponyo who accidentally contributed to this thesis by giving me the bad allergy and those sleepless nights when I could finish up the writing ...